

EXHIBIT 4

**THIS EXHIBIT HAS BEEN
REDACTED IN ITS ENTIRETY**

EXHIBIT 5

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

IN THE UNITED STATES DISTRICT COURT
IN AND FOR THE DISTRICT OF DELAWARE

- - -

ACCELERATION BAY LLC, : CIVIL ACTION
Plaintiff, :
vs. :
ACTIVISION BLIZZARD, INC., :
a Delaware Corporation, :
Defendant. : NO. 15-228 (RGA)

----- :
ACCELERATION BAY LLC, : CIVIL ACTION
Plaintiff, :

vs. :
ELECTRONIC ARTS INC., :
Defendant. : NO. 15-282 (RGA)

----- :
ACCELERATION BAY LLC, : CIVIL ACTION
Plaintiff, :

vs. :
TAKE-TWO INTERACTIVE :
SOFTWARE, INC., ROCKSTAR :
GAMES, INC. and 2K SPORTS, :
INC., :
Defendants. : NO. 15-311 (RGA)

Wilmington, Delaware
Friday, February 12, 2016
3:42 o'clock, p.m.

- - -

BEFORE: HONORABLE RICHARD G. ANDREWS, U.S.D.C.J.

Valerie J. Gunning
Official Court Reporter

1 infringement contentions, so we're now into what happens
2 after he has providing infringement contentions. Right?

3 MR. TOMASULO: Correct. So if the case is as I
4 described it, then we could provide a witness. If the case
5 is about something that they want to explore, then I don't
6 know how to prepare a witness.

7 THE COURT: Well, so --

8 MR. TOMASULO: That gets to the topics.

9 THE COURT: So, all right. And why do you
10 say you've got this great big hurry to get these depositions
11 in?

12 MR. FRANKEL: Yes. I would just like to, you
13 know, briefly respond to these issues.

14 The first point is the hurry is, we have people
15 reviewing source code now without the benefit of documents,
16 and having the depositions will make the source code review
17 more efficient and more effective. Beyond that, we have a
18 number of deadlines that are coming up in the case. The
19 case is a year old at this point, and among those deadlines
20 are picking the ESI search terms for e-mail discovery.
21 Again, we are two-and-a-half months away from starting the
22 claim construction process and then there are other
23 deadlines coming up, and there are a lot of depositions that
24 we need to take.

25 We have given very detailed complaints. They

1 like that.

2 But it does not make sense to me to be reviewing
3 30(b)(6) topics until the case is defined by infringement
4 contentions. So get the infringement contentions in, meet
5 and confer. You know, the defendants will get you witnesses
6 when you have an understanding of what these witnesses are
7 going to be testifying about.

8 MR. TOMASULO: Thank you, your Honor.

9 MR. FRANKEL: Your Honor, if I could just ask
10 one question briefly.

11 The defendants have filed motions for -- have
12 filed petitions for enter partes review.

13 THE COURT: Okay.

14 MR. FRANKEL: And I imagine in the unlikely
15 event that those petitions are granted, they may consider a
16 motion to stay.

17 THE COURT: That's true.

18 MR. FRANKEL: We have been trying --

19 THE COURT: Or I mean I assume that's true. In
20 fact, haven't you filed such a motion already?

21 MR. TOMASULO: No.

22 THE COURT: I thought you said --

23 MR. FRANKEL: No. They have not moved to stay
24 the case.

25 THE COURT: Or maybe you filed a notice.

1 are not the six page pro forma complaints that identify the
2 features. His description of the technology that we're
3 interested in is the technology that we're interested in.
4 That's what we've talked about in our meet and confers.

5 If you take a look at the topics, they are
6 specific and they are directed to those issues. And the one
7 point he mentioned about the situation where there's a
8 network of servers that appears, that is described in the
9 complaint, and I've identified those paragraphs to
10 defendants' counsel.

11 So we think they've been given fair notice of
12 what our theory is and we're going to give our infringement
13 contentions, and we don't want to wait until April or May to
14 start this process.

15 We gave them specific topics --

16 THE COURT: All right.

17 MR. FRANKEL: Yes.

18 THE COURT: Well, so, you get them the
19 infringement contentions on March 2nd. You know, you can't
20 really schedule depositions until you have them. After you
21 get them the infringement contentions, since you both seem
22 to understand what the architecture is, talk to each other.
23 Relate, you know, because I also understand part of their
24 complaint is, you know, your 30(b)(6) topics are too general
25 or too vague, or they don't know what they are, something

1 MR. TOMASULO: We filed a notice. I think we
2 are required to file a notice, so we did do that.

3 THE COURT: Well, good that you did.

4 MR. BLUMENFELD: Some Judges require it.

5 THE COURT: Okay. I don't think I actually do,
6 but I do like to know about such things.

7 But, in any event, you're unlikely to be making
8 any motions until you see how -- until six months go by,
9 more or less, right, and that you either get an inclusion
10 decision or not?

11 MR. TOMASULO: We -- I don't know when we'll
12 make such a motion, but I think what he's going to ask is
13 that if we don't hold it against them that he -- the status
14 of taking depositions.

15 THE COURT: Well, I would say -- you know, I
16 actually --

17 MR. TOMASULO: And I don't have --

18 THE COURT: So just on that topic, I did look at
19 the docket, because I thought part of what you said,
20 Mr. Frankel, was that the defendants had noticed four
21 depositions.

22 MR. FRANKEL: The defendants noticed four
23 depositions, and a month ago when we were here, agreed to
24 proceed with these depositions.

25 THE COURT: Well, but just in terms of noticing

1 likely that they have the design documents that describe the
2 types of networks that their code creates, and that's what
3 we wanted to get from defendants.

4 THE COURT: Okay. Design documents. Does
5 Demonware have design documents?

6 MR. TOMASULO: The way these -- I don't know.
7 The answer is, the way these games are -- these two games
8 were built by two different, they're called studios.

9 THE COURT: Okay.

10 MR. TOMASULO: And so those studios -- for Call
11 of Duty, there have been different studios over the years.
12 And so those -- that studio is an entity unto itself even
13 though those two studios, say like Sledgehammer are owned by
14 Activision.

15 Those studios, we went to their people and we
16 got -- that's how we got the source code in the first place
17 and that's the source code they reviewed.

18 When they say that there haven't been technical
19 documents produced, that's just simply incorrect.

20 THE COURT: And so --

21 MR. TOMASULO: They're --

22 THE COURT: So he's saying there should be
23 design documents for Call of Duty. You are saying, I take
24 it, we've produced what we have and we looked at Demonware
25 for these things, too. Is that right?

1 it.

2 THE COURT: Okay.

3 MR. FRANKEL: If they do have it, we would like
4 it.

5 THE COURT: Okay. So that issue is resolved;
6 right?

7 MR. TOMASULO: As long as it's what I said and
8 not what he said, because what he said is not correct.

9 THE COURT: Well, what I thought he said is,
10 what I thought we agreed on, there are two principles here,
11 one of which is Demonware is a wholly-owned subsidiary, so
12 you do have control and access over whatever it is they have
13 even though they're a separate company.

14 And the second thing is that to, if they have
15 design documents for these things, you're going to find that
16 out and produce them. And that sounded to me like what you
17 agreed to what I said. It didn't sound to me any different
18 than what he said.

19 MR. TOMASULO: I thought -- yes. We'll do what
20 I said, which is to look for the design documents that are
21 relevant to Call of Duty, and we are not withholding
22 anything like that. All of those things, if they were
23 relevant, they would have been at the studio in the first
24 place.

25 THE COURT: Okay.

1 MR. TOMASULO: Well, this is the first I've
2 heard of a specific request that we go look for Demonware's
3 supposedly design documents.

4 Whatever documents the studio --

5 THE COURT: So let's take care of that.

6 MR. TOMASULO: Okay.

7 THE COURT: Because you're willing to do it. So
8 can you contact Demonware in the next ten days and find out
9 whether they have any design documents, advise Mr. Frankel
10 if they do, and then promptly get them if they do. And if
11 they don't, advise them of that.

12 MR. TOMASULO: So there's a specific type of
13 document called a technical design document. He's a
14 software engineer, so he probably knows what that is. All
15 we can ask Demonware, if they have any CDDs that were
16 relevant, or CDs or something similar that were relevant to
17 how Call of Duty operates.

18 THE COURT: Okay.

19 MR. FRANKEL: Your Honor, this is exactly what
20 we clearly asked for in correspondence and in the meet and
21 confer. Demonware, as a wholly-owned subsidiary, be a part
22 of the discovery process, including core technical
23 discovery, and if it comes down to it, depositions. If
24 all -- and I think counsel has agreed that we will get that
25 discovery from them. If they don't have it, they don't have

1 MR. TOMASULO: But we'll also ask Demonware and
2 we'll do what -- we'll ask Demonware if they have these
3 kinds of documents that are relevant to Call of Duty.

4 THE COURT: Okay. All right.

5 MR. TOMASULO: And we'll produce them if we can
6 find them.

7 THE COURT: Okay.

8 MR. FRANKEL: Your Honor, I think we are close,
9 but I just want to confirm that the fact that Demonware is a
10 subsidiary is not going to be a basis for them to be less
11 involved in the discovery here, because it's -- you know,
12 it's not just for Call of Duty. They designed a kit.

13 THE COURT: Well, you know, we're not talking
14 about discovery generally. We're talking about core
15 technical documents, and for core technical documents,
16 they're a wholly-owned subsidiary.

17 Activision, if they're the one who have the
18 relevant technical documents, Activision needs to get them
19 from Demonware, and I think twice now, Mr. Tomasulo has said
20 he would.

21 MR. TOMASULO: What I want -- there is a
22 distinction here, and what -- Demonware is not an accused
23 product. They have more -- so they have some files that are
24 incorporated into the client side server, into the client
25 side code for Call of Duty. That was produced, and we'll go

EXHIBIT 6

JS-6

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

Priority _____
Send _____
Enter _____
Closed _____
JS-5/JS-6 _____
Scan Only _____

CIVIL MINUTES - GENERAL

CASE NO.: CV 16-00737 SJO (MRWx) DATE: July 25, 2016

TITLE: Activision Publishing, Inc. v. xTV Networks LTD., et al.

=====
PRESENT: THE HONORABLE S. JAMES OTERO, UNITED STATES DISTRICT JUDGE

Victor Paul Cruz Not Present
Courtroom Clerk Court Reporter

COUNSEL PRESENT FOR PLAINTIFF: COUNSEL PRESENT FOR DEFENDANTS:

Not Present Not Present

=====
PROCEEDINGS (in chambers): ORDER GRANTING DEFENDANTS' MOTION TO DISMISS PLAINTIFF'S FIRST AMENDED COMPLAINT [Docket No. 27]

This matter is before the Court on Defendants xTV Networks, Ltd. and xTV Networks US, Inc.'s., (together, "Defendants")¹ Motion to Dismiss Plaintiff's First Amended Complaint ("Motion"), filed April 26, 2016. Plaintiff Activision Publishing, Inc. ("Plaintiff") opposed the Motion ("Opposition") on May 10, 2016, and Defendants replied ("Reply") on May 17, 2016. On May 16, 2016, the Court invited the parties to file supplemental briefs in light of new authority from the United States Court of Appeals for the Federal Circuit. Defendants filed their supplemental brief ("Defendants' Supplemental Brief") on June 6, 2016, and Plaintiff filed its supplemental brief ("Plaintiff's Supplemental Brief") on June 13, 2016. The Court found this matter suitable for disposition without oral argument and vacated the hearing set for May 23, 2016. See Fed. R. Civ. P. 78(b). For the reasons stated below, the Court **GRANTS** Defendants' Motion.

I. FACTUAL AND PROCEDURAL HISTORY

Plaintiff commenced the instant patent litigation against Defendants on February 2, 2016 and filed its First Amended Complaint for Patent Infringement ("FAC") on April 11, 2016. (See Compl., ECF No. 1; FAC, ECF No. 22.) In its FAC, Plaintiff alleges that Defendants directly infringe U.S. Patent No. 6,549,933 ("the '933 Patent") through their manufacture, use, sale, importation, licensing and/or offering for sale of the following products: "xTV Now," "Digital Signage," "Compute Stick," and related software and services ("Accused Products"). (FAC ¶¶ 13, 31-35.) Plaintiff further alleges that Defendants indirectly infringe the '933 Patent by contributing to and actively inducing others, including end user customers, to directly infringe the '933 patent by their use, sell, import, and/or offer for sale of the Accused Products. (FAC ¶¶ 36-42.)

¹ Previously named defendant xTVNow, Inc. was dismissed from this action without prejudice pursuant to Federal Rule of Civil Procedure 41(a)(1) on March 8, 2016. (See Notice of Dismissal as to xTVNow, Inc., ECF No. 15.)

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

CIVIL MINUTES - GENERAL

CASE NO.: CV 16-00737 SJO (MRWx)

DATE: July 25, 2016

The '933 Patent, titled "Managing, Accessing, and Retrieving Networked Information Using Physical Objects Associated with the Networked Information," issued on April 15, 2003, and the application leading to the '933 Patent was filed on August 4, 1998. (FAC ¶ 30, Ex. K ("933 Patent").) The '933 Patent claims network-based systems for managing information through the use of independently portable physical objects. (See '933 Patent cols. 13-18.) The '933 Patent contains three independent claims—claims 1, 28, and 55—and seventy-seven (77) dependent claims. (*Id.*) Plaintiff alleges that Defendants infringe at least claims 28 and 55, and Plaintiff does not appear to challenge this assertion in its Opposition. (FAC ¶¶ 31-33, 38-40; Opp'n 8-15, ECF No. 28.) Claims 28 and 55 read in their entirety:

28. A **method of information management** in a network-based system, comprising the steps of:
- (a) reading an ID (referred to as a Thing ID) from an independently portable physical object that is a data storage device (referred to as an Informative Thing), wherein:
 - (i) the Thing ID represents an identity fo the Informative Thing; and
 - (ii) the Thing ID is associated with information stored at an information store separate from the Informative Thing;
 - (b) determining a location for where the information is stored based on the Thing ID; and
 - (c) retrieving the information associated with the Informative Thing form the network-based system using the Thing ID.

('933 Patent col. 14 l. 62-col. 15 l. 10 [emphasis added].)

55. An **article of manufacture** comprising an independently portable physical object that is a data storage device (referred to as an Informative Thing), wherein;
- (a) the Informative Thing stores and ID (referred to as a Thing ID), thereon, wherein the Thing ID represents an identity of the Informative Thing;
 - (b) the Thing ID is associated with information stored at an information store separate from the Informative Thing on a network-based system; and
 - (c) the Thing ID is used to determine a location for where the information is stored and retrieve the information associated with the Informative Thing from the information store.

('933 Patent col. 16 ll. 34-47 [emphasis added].)

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

CIVIL MINUTES - GENERAL

CASE NO.: CV 16-00737 SJO (MRWx)

DATE: July 25, 2016

On September 3, 2015, Plaintiff notified Defendants' Managing Director, Joe Ward, of the '933 Patent and offered to license the '933 Patent in connection with the Accused Products. (FAC ¶ 37.) Defendants declined to license rights to the '933 Patent. (FAC ¶ 37.)

II. DISCUSSION

In their Motion, Defendants principally argue that the claims of the '933 Patent are not eligible for patent protection under 35 U.S.C. § 101 ("Section 101") because the methods and systems claimed therein "do nothing more than claim a long-practiced, well known process, and apply it to a computer." (Mot. 1, ECF No. 27-1.) Specifically, Defendants argue that the claims are directed to the abstract concept of "information management," i.e. "organizing data to be able to locate information at a second location," and that the claims contain generic computer hardware that cannot supply the "inventive concept" sufficient to transform this abstract idea into a patent-eligible invention. (See Mot. 13-18.) Defendants also argue that Plaintiff's claims against xTV Networks, Ltd. should be dismissed under Rule 12(b)(2) of the Federal Rules of Civil Procedure ("Rule 12(b)(2)") for lack of personal jurisdiction. (Mot. 2, 19-20.)

Plaintiff responds that the '933 Patent claims a specialized and novel physical, portable device for the retrieval of information stored in a computer network providing limited access through a specific retrieval process, and does not broadly encompass "storing an identification at one location and using that identification to locate and retrieve something at a second location." (See Opp'n 1, ECF No. 29.) Plaintiff thus argues that the invention underlying the claims of the '933 Patent has "features unique to computing which improve the functioning of the computers themselves and which address a technological limit with respect to data storage in network-based computer systems that existed [in 1998]." (Opp'n 1.) Plaintiff also argues that the claim terms "Thing ID" and "Informative Thing" have specific definitions set forth in the specification of the '933 Patent, and that it would be improper to invalidate the asserted claims prior to claim construction. (Opp'n 2.)

With the parties' arguments laid out, the Court now examines the applicable legal standards.

A. Section 101 Analytical Framework

"Section 101 defines the subject matter that may be patented under the Patent Act." *Bilski v. Kappos*, 561 U.S. 593, 601 (2010). Section 101 reads in its entirety: "Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title." 35 U.S.C. § 101. "Section 101 thus specifies four independent categories of inventions or discoveries that are eligible for patent protection: processes, machines, manufactures, and compositions of matter." *Bilski*, 561 U.S. at 601.

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

CIVIL MINUTES - GENERAL

CASE NO.: CV 16-00737 SJO (MRWx)

DATE: July 25, 2016

Although acknowledging that "[i]n choosing such expansive terms . . . Congress plainly contemplated that the patent laws would be given wide scope," the Supreme Court long ago identified three exceptions to Section 101: "laws of nature, physical phenomena, and abstract ideas." *Diamond v. Chakrabarty*, 447 U.S. 303, 308-09 (1980). Although these exceptions are not required by the statutory text, they are consistent with the idea that certain discoveries "are part of the storehouse of knowledge of all men" and are "free to all men and reserved exclusively to none." *Funk Bros. Seed Co. v. Kalo Inoculant Co.*, 333 U.S. 127, 130 (1948). Thus, "the concern that drives this exclusionary principle [is] one of pre-emption." *Alice Corp. Pty. v. CLS Bank Int'l*, 134 S. Ct. 2347, 2354 (2014) (citation omitted). Consequently, the Supreme Court has required that "[i]f there is to be invention from such a discovery, it must come from the application of the law of nature to a new and useful end." *Funk Bros.*, 333 U.S. at 130. These principles have been held to apply with equal force to product and process claims. *Gottschalk v. Benson*, 409 U.S. 63, 67-68 (1972).

Alice Corp. v. CLS Bank ("*Alice*") represents the Supreme Court's latest attempt to clarify how courts should apply these difficult principles. In *Alice*, the Supreme Court expanded on the two-step approach for resolving Section 101 issues first articulated in *Mayo Collaborative Servs. v. Prometheus Labs., Inc.*, 132 S. Ct. 1289, 1296-97 (2012). First, a court must "determine whether the claims at issue are directed to one of those patent-ineligible concepts." *Alice*, 134 S. Ct. at 2355 (citing *Mayo*, 132 S. Ct. at 1296-97). If so, then the court must ask "[w]hat else is there in the claims," which requires consideration of "the elements of each claim both individually and 'as an ordered combination' to determine whether the additional elements 'transform the nature of the claim' into a patent-eligible application." *Id.* (citing *Mayo*, 132 S. Ct. at 1297-98). In this second step, the court must "search for an 'inventive concept'—*i.e.*, an element or combination of elements that is 'sufficient to ensure that the patent in practice amounts to significantly more than a patent upon the [ineligible concept] itself.'" *Id.* (citing *Mayo*, 132 S. Ct. at 1294). This two-step analytical framework has been labeled the "*Alice/Mayo* test" or simply the "*Alice* test."

Identifying whether a claim is "directed to an abstract idea" under step one of the *Alice/Mayo* test is not always a simple undertaking. Although there is some disagreement among courts as to how expansively a claim should be examined at *Alice/Mayo* step one, the Federal Circuit recently instructed that the "'directed to' inquiry applies a stage-one filter to claims, considered in light of the specification, based on whether 'their character as a whole is directed to excluded subject matter.'" *Enfish, LLC v. Microsoft Corp.*, — F.3d —, 2016 WL 2756255, at *4 (Fed. Cir. May 12, 2016) (quoting *Internet Patents Corp. v. Active Network, Inc.*, 790 F.3d 1343, 1346 (Fed. Cir. 2015)); see also *Cal. Inst. Tech. v. Hughes Commc'ns, Inc.*, 59 F. Supp. 3d 974, 991-92 (C.D. Cal. 2014) (requiring that a court "identify the purpose of the claim—in other words, what the claimed invention is trying to achieve—and ask whether that purpose is abstract," making the *Alice/Mayo* step 1 "a sort of 'quick look' test, the object of which is to identify a risk of preemption and ineligibility"); *DDR Holdings, LLC v. Hotels.com, L.P.*, 773 F.3d 1245, 1258-59 (Fed. Cir. 2014) (although blurring steps one and two in analyzing internet-based patent claims, finding the claims not patent-ineligible where they "specify how interaction with the Internet are manipulated to yield

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

CIVIL MINUTES - GENERAL

CASE NO.: CV 16-00737 SJO (MRWx)

DATE: July 25, 2016

a desired result—a result that overrides the routine and conventional sequence of events ordinarily triggered by the click of the hyperlink"). It is "relevant to ask whether the claims are directed to an improvement to computer functionality versus being directed to an abstract idea, even at the first step of the *Alice* analysis," which can entail reviewing both the claim language and the specification. *Enfish*, — F.3d —, 2016 WL 2756255, at *4-*6.

If the claim is directed to an abstract idea, the Court must then determine whether the specific claim elements, considered both individually and "as an ordered combination," "transform the nature of the claim" into a patent-eligible invention. *Alice*, 134 S. Ct. at 2355 (quoting *Mayo*, 132 S. Ct. at 1297). In *Alice*, the Supreme Court considered whether "[t]he introduction of a computer into the claims" directed toward the abstract idea of intermediated settlement was sufficient to "transform the nature of the claim" by adding an "inventive concept." *Id.* at 2357. The Supreme Court held that it did not, and made clear that "the mere recitation of a generic computer cannot transform a patent-ineligible abstract idea into a patent-eligible invention." *Id.* at 2358. "Nor is limiting the use of an abstract idea 'to a particular technological environment'" sufficient to impart patent-eligibility. *Id.* (quoting *Bilski*, 561 U.S. at 610-11). In its discussion, the Supreme Court in *Alice* distinguished an earlier case, *Diamond v. Diehr*, in which the Court held that a computer-implemented process for curing rubber, which employed a "well-known" mathematical equation, was nevertheless patent-eligible because it used that equation in a process designed to solve a technological problem in "conventional industry practice."² *Alice*, 134 S. Ct. at 2358 (citing *Diehr*, 450 U.S. at 177-78). Moreover, the Federal Circuit recently held in *Bascom Global Internet Services, Inc. v. AT&T Mobility LLC* that "an inventive concept can be found in the non-conventional and non-generic arrangement of known, conventional pieces," particularly where the invention claimed is more than the implementation of an abstract idea "on generic computer components, without providing a specific technical solution beyond simply using generic computer concepts in a conventional way." — F.3d —, 2016 WL 3514158, at *6, *8 (Fed. Cir. June 27, 2016).

With this high-level understanding of the purpose and limits of Section 101, the Court addresses whether a motion to dismiss may properly be brought on Section 101 grounds.

B. The Appropriateness of Ruling on Section 101 Motions at the Pleadings Stage and the Defendant's Burden

Federal Rule of Civil Procedure 12(b)(6) permits a party to move to dismiss an action for "failure to state a claim upon which relief can be granted" if "made before pleading if a responsive pleading is allowed." Fed. R. Civ. P. 12(b)(6). "Patent eligibility under [Section] 101 is a question

² In particular, the Supreme Court in *Diehr* explained that the claimed contribution to the art was the step of "constantly measuring the actual temperature inside a rubber molding press" used to create synthetic rubber products. *Diehr*, 450 U.S. at 206.

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

CIVIL MINUTES - GENERAL

CASE NO.: CV 16-00737 SJO (MRWx)

DATE: July 25, 2016

of law that may, in appropriate cases, be decided on the pleadings without the benefit of a claim construction hearing." *Modern Telecom Sys. LLC v. Earthlink, Inc.*, No. CV 14-0347 DOC, 2015 WL 1239992, at *6 (C.D. Cal. Mar. 17, 2015) (citing *Content Extraction & Transmission LLC v. Wells Fargo Bank, Nat. Ass'n*, 776 F.3d 1343, 1349 (Fed. Cir. 2014) (affirming district court's decision to grant motion to dismiss based on patent-ineligible subject matter under Section 101 without having a claim construction hearing); *Ultramercial, Inc. v. Hulu, LLC* ("*Ultramercial II*"), 772 F.3d 709, 711 (same). However, "it will ordinarily be desirable—and often necessary—to resolve claim construction disputes prior to a [Section] 101 analysis, for the determination of patent eligibility requires a full understanding of the basic character of the claimed subject matter." *Bancorp Servs., L.L.C. v. Sun Life Assurance. Co. Can. (U.S.)*, 687 F.3d 1266, 1273-74 (Fed. Cir. 2012); *but see Content Extraction*, 776 F.3d at 1349 ("Although the determination of patent eligibility requires a full understanding of the basic character of the claimed subject matter, claim construction is not an inviolable prerequisite to a validity determination under [Section] 101.").

"Although the clear and convincing evidence standard is not applicable" to a Section 101 motion brought pursuant to Federal Rule of Civil Procedure 12(b)(6), the movant "still bear[s] the burden of establishing that the claims are patent-ineligible under [Section] 101." *Modern Telecom*, 2015 WL 1239992 at *8. "Additionally, in applying [Section] 101 jurisprudence at the pleading stage, the Court construes the patent claims in a manner most favorable to Plaintiff." *Id.* (citing *Content Extraction*, 776 F.3d at 1349).

C. Analysis

Having determined that the Court can rule on Defendants' Motion at the pleadings stage, the Court now applies the *Alice/Mayo* test to determine whether the claims of the '933 Patent are patent-eligible. Before diving into step one, however, the Court notes that Defendants' arguments principally relate to patent-eligibility of claim 28 of the '933 Patent, which Defendants argue is "representative" of the rest of the claims in the '933 Patent for the purposes of the Section 101 analysis. (Mot. 11.) Plaintiff does not dispute the representativeness of claim 28, (*see generally* Opp'n), and the Court independently finds the claims sufficiently similar to analyze them together for Section 101 purposes.

1. Step 1: The Claims Are Directed to a Patent-Ineligible Abstract Idea

The first step in determining whether a claim satisfies Section 101 is determining whether the claim at issue is directed to a "patent-ineligible concept" such as an abstract idea. *Alice*, 134 S. Ct. at 2355. Under the first step of the *Alice/Mayo* test, "the court must identify the purpose of the claim—in other words, what the claimed invention is trying to achieve—and ask whether that purpose is abstract." *Cal. Inst. Tech.*, 59 F. Supp. 3d at 991. To accomplish this task, the Court must "appl[y] a stage-one filter to claims, considered in light fo the specification, based on whether 'their character as a whole is directed to excluded subject matter.'" *Enfish*, — F.3d —, 2016 WL 2756255, at *4 (quoting *Internet Patents Corp.*, 790 F.3d at 1346). Moreover, because "all

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

CIVIL MINUTES - GENERAL

CASE NO.: CV 16-00737 SJO (MRWx)

DATE: July 25, 2016

improvements in computer-related technology are inherently abstract," it is "relevant to ask whether the claims are directed to an improvement to computer functionality versus being directed to an abstract idea, even at the first step of the *Alice* analysis." *Id.*

Defendants in their Motion argue that claims 28 and 55 of the '933 Patent are directed to "the general abstract idea of storing an identification ("Thing ID") on a portable device ("Informative Thing") and using that ID to locate and retrieve something associated with the ID at a second location." (Mot. 13-14.) Thus, according to Defendants, "[c]laim 28 broadly recites a 'method of information management,' and the claim elements recite nothing more than using an ID stored on a portable device to retrieve information stored elsewhere (e.g., in a network)." (Mot. 14.) Plaintiff responds that the '933 Patent "seeks to deviate from [] broad and traditional information management and claim a specialized portable device used to improve the functioning of networked computers and provide end users with something tangible that they can easily use and understand to manage their information while limiting access." (Opp'n 9-10.) Plaintiff additionally argues that Defendants' proposed "abstract idea" improperly "'strip[s] away' the claim elements that recite hardware," and overlooks that the invention underlying the asserted claims requires a specialized device and a network-based computer system. (Opp'n 10-11.)

In light of the Federal Circuit's decisions in *Enfish* and *In re TLI Communications LLC Patent Litigation*, — F.3d —, 2016 WL 2865693 (Fed. Cir. May 17, 2016), Defendants argue that the question the Federal Circuit now requires courts to ask is whether inventors chose to solve the problems facing them by using "conventional or generic technology," or instead by "building a better computer." (Defs.' Supp'l Br. 2, ECF No. 32.) As Defendants would have it, the claims of the '933 Patent "are not directed to a specific improvement to computer functionality. Rather, they are directed to the use of conventional or generic technology in a nascent but well-known environment,' without any claim that the invention reflects a better computer." (Defs.' Supp'l Br. 2 (quoting *TLI*, — F.3d —, 2016 WL 2865693, at *3).) Defendants also cite a post-*Enfish* case from this district finding the claims asserted therein patent-ineligible—*Kinglite Holdings Inc. v. Micro-Star Int'l Co.*, No. CV 14-03009 JVS (PJWx) (C.D. Cal. May 26, 2016), ECF No. 226—that they contend involves "remarkably similar" claims to the '933 Patent's claims. (Defs.' Supp'l Br. 4.)

Plaintiff responds that the '933 Patent, like the patent at issue in *Enfish*, is directed to specific improvements in the functioning of computers. (See Pl.'s Supp'l Br., ECF No. 33.) Plaintiff points to language in the abstract noting that in 1998, data was stored traditionally in one of three basic forms—(1) locally on a portable object (e.g. a floppy disc); (2) on a local computer memory; or (3) remotely on a network for access by a computer. (Pl.'s Supp'l Br. 3 [citing '933 Patent cols. 1 ll. 14-16, 2 ll. 8-30].) Plaintiff argues that the '933 Patent deviates from these prior approaches "by combining a portable, physical device having an identification with a remote computer network (that includes a look-up table)." (Pl.'s Supp'l Br. 3.) Plaintiff contends that the '933 Patent teaches that the following benefits and improvements in the way that computers operate: (1) improved security, ('933 Patent col. 6 l. 25-col. 7 l. 24, col. 9 ll. 41-57); (2) improved simplicity, ('933 Patent

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

CIVIL MINUTES - GENERAL

CASE NO.: CV 16-00737 SJO (MRWx)

DATE: July 25, 2016

col. 2 ll. 14-30, col. 3 ll. 1-5, col. 10 ll. 49-67); (3) improved file sharing over a network, ('933 Patent col. 2 ll. 54-60); and (4) increased storage capacity, ('933 Patent col. 2 ll. 31-35).

According to the specification of the '933 Patent, the general purpose of the claimed invention is to provide a "new approach[] to information management . . . that simplif[ies] the steps a user must go through to copy a file from one physical location to another on a ftp site or on the web, as well as the time involved to perform those steps." ('933 Patent col. 1 ll. 29-49.) The invention claimed in the '933 Patent purports to "relieve[] the user of many burdensome tasks involved in naming, finding, saving, and managing networked information by providing a simple, physical user interface to the information." ('933 Patent col. 2 ll. 37-41.) In one preferred embodiment, a physical object, such as a floppy disk, has a "unique identifier" that locates and retrieves data stored on the network and transfers this data to the local system when the floppy disk is placed on the hard drive. ('933 Patent col. 2 ll. 42-53.) Later in the specification, it is revealed that a tertiary purpose of the invention is to enable the transfer of data in a simple and fast manner that is "not limited by floppy disk capacity" because the data is stored on a separate server. ('933 Patent col. 7 ll. 45-62.)

With this understanding of the stated goals of the '933 Patent, the Court finds that the claims of the '933 Patent are directed to the abstract idea of using information stored in one place to determine the location of and retrieve information stored in a second place, i.e., a form of "information management." This abstract idea underlying the claimed systems, methods, and articles of manufacture, which is similar to the characterization urged by Defendants, relies on evidence intrinsic to the '933 Patent and better characterizes the nature and purpose of the asserted claims. This finding is further supported by looking to the name of the '933 Patent: "Managing, Accessing, and Retrieving Networked Information Using Physical Objects Associated with the Networked Information."

A review of the independent claims reveals that they are directed to the concept of managing and retrieving data on a network-based system. Claim 28, for example, recites a "method of information management in a network-based system," and further requires (1) reading an ID ("Thing ID") from an independently portable physical object that is a data storage device ("Informative Thing"), where the Thing ID represents an identify of the Informative Thing and the Thing ID is associated with information stored at an information store separate from the Informative Thing; (2) "determining a location for where the information is stored based on the Thing ID"; and (3) "retrieving the information associated with the Informative Thing from the network-based system using the Thing ID." ('933 Patent col. 14 l. 62-col. 15 l. 10.) Claim 55 similarly recites an article of manufacture comprising an Informative Thing wherein "the Thing ID represents an identity of the Informative Thing[,]" "the Thing ID is associated with information stored at an information store separate from the Informative Thing on a network-based system[,]" and "the Thing ID is used to determine a location for where the information is stored and retrieve the information associated with the Informative Thing from the information store." ('933 Patent col. 16 ll. 34-47.) There can be no dispute that the purpose of these claims is to manage and retrieve

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

CIVIL MINUTES - GENERAL

CASE NO.: CV 16-00737 SJO (MRWx)

DATE: July 25, 2016

information on a network-based system, and generic reference to technological components is largely irrelevant to the analysis of *Alice/Mayo* step 1. See *Cal. Inst. of Tech.*, 59 F. Supp. 3d at 993 ("Courts must ignore generic recitation of hardware at step one, when the claimed hardware essentially performs a method."); *Intellectual Ventures I LLC v. Capital One Bank (USA)*, 792 F.3d 1363, 1366 (Fed. Cir. 2015) ("An abstract idea does not become nonabstract by limiting the invention to a particular field of use or technological environment, such as the Internet.").

The specification confirms the abstract goals of the '933 Patent. Because claims "must be read in view of the specification, of which they are a part," the Court considers these descriptions of the inventions claimed in the '933 Patent instructive. *IPLearn-Focus, LLC v. Microsoft Corp.*, No. 14-cv-00151-JD, 2015 WL 4192092, at *4 (N.D. Cal. July 10, 2015) (quoting *Phillips v. AWH Corp.*, 415 F.3d 1303, 1315 (Fed. Cir. 2005) (en banc)). The specification describes the invention as "relat[ing] in general to information management, and more particularly, to managing networked information using physical objects." ('933 Patent col. 1 ll. 8-12.) The specification further describes the invention as "a method for labeling physical objects and for associating those labels with network information sources." ('933 Patent col. 2 ll. 35-37.) Indeed, the specification provides that "[t]he Informative Things concept is broadly applicable, as it provides a **general method for managing and transferring information.**" ('933 Patent col. 7 ll. 41-43 [emphasis added].) More particularly, the specification describes the "Informative Thing" as "a physical object [that] stores a pointer to the desired information," which "[u]pon interfacing to a network-based system, . . . is retrieved and then used to retrieve the desired information associated with the Informative Thing from a network-based system." ('933 Patent col. 1 ll. 44-50.) Thus, here, as in *TLI*, "[t]he specification does not describe a new [computer or portable device], a new server, or a new physical combination of the two." *TLI*, — F.3d —, 2016 WL 2865693, at *3. Moreover, "[t]he specification fails to provide any technical details for the tangible components, but instead predominantly describes the system and methods in purely functional terms." *Id.* Viewed in light of the specification, the claims of the '933 Patent are far more readily characterized as being directed to an abstract concept than those in *DDR* and *Enfish*.

The concept of managing and retrieving information by storing an identification on a portable device and using that identification to retrieve information at a second location is similar to other data management concepts found by courts to constitute abstract ideas. See *Cogent Med. Inc. v. Elsevier Inc.*, 70 F. Supp. 3d 1058, 1064-66 (N.D. Cal. Sept. 30, 2014) (granting motion to dismiss for failure to claim patentable subject matter where the claims described a database of medical resources searchable through a library interface); see also *Digitech Image Techs., LLC v. Electronics for Imaging, Inc.*, 758 F.3d 1344, 1351 (Fed. Cir. 2014) (upholding finding of patent-ineligibility where claims "recite[] a process of taking two data sets and combining them into a single data set, the device profile," where the "two data sets are generated by taking existing information . . . and organizing this information into a new form" without requiring input from a physical device). Moreover, the Court agrees with Defendants that the concept of using a "pointer" found in one place to locate and retrieve information found in a separate place is indistinguishable from the idea of using an information desk, a catalogue, or the Dewey Decimal

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

CIVIL MINUTES - GENERAL

CASE NO.: CV 16-00737 SJO (MRWx)

DATE: July 25, 2016

System to locate a desired library book, either within the same library or in a separate library using, say, an interlibrary loan.³ This concept is far from new. See *Cloud Satchel, LLC v. Amazon.com, Inc.*, 76 F. Supp. 3d 553, 562-64 (D. Del. 2014) (claims were directed to the "abstract idea of cataloguing documents to facilitate their retrieval from storage in the field of remote computing"), *aff'd mem.*, 626 Fed. App'x 1010 (Fed. Cir. 2015), *cert. denied*, 2016 WL 1059941 (Apr. 25, 2016).

The Court is not persuaded by Plaintiff's argument that the claims of the '933 Patent are "directed to a specific implementation of a solution to a problem in the software arts," and instead finds that the claims "recite[] generalized steps to be performed on a computer using conventional computer activity . . ." *Enfish*, — F.3d —, 2016 WL 2756255, at *7. The claims of the '933 Patent are readily distinguishable from those at issue in *Enfish* and *DDR Holdings, LLC v. Hotels.com, L.P.*, the sole post-*Alice* Federal Circuit opinions that have found software claims to pass muster under *Alice* step one. In *Enfish*, the asserted claims were specifically directed to a "self-referential table for a computer database," which, according to the specification, functions differently than conventional database structures. See *Enfish*, — F.3d —, 2016 WL 2756255, at *5-*7. In *DDR*, the asserted claims "specif[ied] how interactions with the Internet are manipulated to yield a desired result—a result that overrides the routine and conventional sequence of events ordinarily triggered by the click of a hyperlink." 773 F.3d 1245, 1258 (Fed. Cir. 2014). Thus, the inventions underlying the claims at issue in both cases altered the ordinary functioning of the computing process, yielding "increased flexibility, faster search times, and smaller memory requirements" in *Enfish* and automating the creation of a composite web page by an "outside provider" that incorporates elements from multiple sources in *DDR*. *Enfish*, — F.3d —, 2016 WL 2756255, at *6; *DDR*, 773 F.3d at 1259.

Here, by contrast, the claims of the '933 Patent do **not** purport to alter the structure or function of the physical object, the computer, or indeed any part of the network of computers. Instead, the claims specify that the "physical object that is a data storage device," referred to as an "Informative Thing," "stores an ID (referred to as a Thing ID)," which "represents an identity of the Informative Thing." ('933 Patent col. 16 ll. 35-40.) The claims further require that the Thing ID both (1) be "associated with information stored at an information store separate from the Informative Thing on a network-based system;" and (2) be "used to determine a location for where the information is stored and retrieve the information associated with the Informative Thing from the information store." ('933 Patent col. 16 ll. 41-48.) None of these elements recite specific hardware or software, and the specification makes clear that "the present invention may be implemented as a method, apparatus, or article of manufacture using **standard programming**

³ Plaintiff's argument that the '933 Patent "has no application at all to the retrieval of physical items," but instead "is directed to the retrieval of digital information" does not persuade, for "a claim reciting an abstract idea does not become eligible merely by adding the words "apply it" on a computer. *Bancorp Servs.*, 687 F.3d at 1276.

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

CIVIL MINUTES - GENERAL

CASE NO.: CV 16-00737 SJO (MRWx)

DATE: July 25, 2016

and/or engineering techniques to produce software, firmware, hardware, or any combination thereof." ('933 Patent col. 4 ll. 1-4 [emphasis added].) Indeed, the preferred "hardware environment" includes networked computers that include a "processor, random access memory (RAM), data storage devices, display devices, input devices, etc." ('933 Patent col. 3 ll. 33-50.)

Accordingly, the Court finds that the claims of the '933 Patent are directed to the abstract idea of using information stored in one place to determine the location of and retrieve information stored in a second place. The Court now turns to the second prong of the patent-eligibility inquiry.

2. Step 2: Whether Claims 28 and 55 Include an "Inventive Concept" Sufficient to "Transform the Nature of the Claim[s]" into Patentable Inventions

Having determined that claims 28 and 55 of the '933 Patent are directed to the abstract idea of using information stored in one place to determine the location of and retrieve information stored in a second place, the Court now "examine[s] the limitations of the claims to determine whether the claims contain an 'inventive concept' to 'transform' the claimed abstract idea into patent-eligible subject matter." *Ultramercial II*, 772 F.3d at 715 (quoting *Alice*, 134 S. Ct. at 2357). "A claim that recites an abstract idea must include 'additional features' to ensure 'that the [claim] is more than a drafting effort designed to monopolize the [abstract idea].'" *Alice*, 134 S. Ct. at 2357 (quoting *Mayo*, 132 S. Ct. at 1297) (alterations in original). "Those 'additional features' must be more than 'well-understood, routine, conventional activity.'" *Ultramercial II*, 772 F.3d at 715 (quoting *Mayo*, 132 S. Ct. at 1298). The "mere recitation of a generic computer cannot transform a patent-ineligible abstract idea into a patent-eligible invention." *Alice*, 134 S. Ct. at 2358.

Plaintiff contends that the recitation of three claim terms—"Informative Thing," "Thing ID," and "information store separate from the Informative Thing on a network-based system"—in claims 28 and 55 demonstrate that these claims are "directed to a specialized portable device for a specific form of information management in network computers," or at the very least, demonstrate that claim construction is required before the Court can rule on the Section 101 issue. (See Opp'n 12-14.) Defendants argue instead that the claims are "sufficiently straightforward" that formal claim construction is not necessary to understand their content, and therefore do not proffer their preferred constructions of these terms. (Mot. 12.)

The Court agrees with Defendants that claim construction is not necessary to answer the Section 101 question, and concludes that claims 28 and 55 do not contain sufficient "additional features" to ensure that these claims are "more than a drafting effort designed to monopolize the [abstract idea]" to which they are directed. Although the Court does not endeavor to construe the claims of the '933 Patent at this juncture, it notes that claim 28 recites "[a] network-based system" in the preamble and contains limitations involving a "portable physical object that is a data storage device[.]" ('933 Patent col. 14 l. 62-col. 15 l. 10.) Similar to the system claims at issue in *Alice*, this method claim recites hardware that appears on its face to be "purely functional and generic," such as the "data processing system" with a "communications controller" and "data storage unit"

13-8
UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

CIVIL MINUTES - GENERAL

CASE NO.: CV 16-00737 SJO (MRWx)

DATE: July 25, 2016

that are found in "[n]early every computer." *Alice*, 134 S. Ct. at 2360. Indeed, the majority of the steps of the independent claims at issue do little more than recite generic computer hardware necessary to perform the abstract concept of managing and retrieving information associated with an ID. (See '933 Patent col. 14 l. 62-col. 15 l. 10.) "Adding routine additional steps," such as reading an ID and determining a location for where the information is stored based on an ID, "does not transform an otherwise abstract idea into patent-eligible subject matter." See *Ultramercial II*, 772 F.3d at 716. Additionally, the invocation of a network and recitation of routine and generic processing from a "network-based system" are not inventive concepts. Tellingly, the specification of the '933 Patent describes that the general idea underlying the '933 Patent's approach, that physical objects can be associated with an "identifier [that] serves as the key in a lookup table associating a physical thing with information," "is not new." ('933 Patent col. 3:7-12.) Beyond the abstract idea of managing data in a network-based system, the claims of the '933 Patent merely recite "well-understood, routine conventional activit[ies]" either by requiring conventional computer activities or routine data gathering steps. See *Alice*, 134 S. Ct. at 2357 (quoting *Mayo*, 132 S. Ct. at 1294, 1298). The claims here do not "do more than simply instruct the practitioner to implement the abstract idea of" managing and retrieving information "on a generic computer." See *Alice*, 134 S. Ct. at 2347, 2359.

Although Plaintiff throughout its Opposition characterizes the "Informative Thing" as a "specialized physical object [that is] used to improve the functioning of networked computers," (see Opp'n 9, 12), none of the claims of the '933 Patent recite any "specialized physical object;" instead, the claims themselves broadly define the term Informative Thing as "an independently portable physical object that is a data storage device," (see '933 Patent cols. 14 ll. 64-67, 16 ll. 35-37). Thus, the claims do not contain any meaningful limits as to this claim element. The Court does not find construction of this claim term to be necessary, and Plaintiff does not proffer its preferred construction as to this term or any other term. See *IPLearn-Focus*, 2015 WL 4192092, at *5-6 (finding patent-ineligible claims that "fail to recite or disclose any non-routine or unconventional method for solving a uniquely Internet-based problem" and noting that the patentee was unable to "identify any way in which the claims 'purport to improve the functioning of the computer itself or 'effect an improvement in any other technology or technological field'"). To the extent that claim construction could be aided by referring to the specification, such reference would not be of any assistance to Plaintiff, as the specification reveals that the "Informative Thing" can be any number of devices, including a "floppy disk, CD-ROM, magneto-optical disk, memory card, etc." or a device that "can be accessed with specialized hardware." ('933 Patent col. 4 ll. 31-50.) The recitation of a generic computer accessory such as a "data storage device" cannot supply an "inventive concept" sufficient to render the claims patent-eligible under Section 101.

Similarly, the recitation of a "Thing ID," which "**represents an identity** of the Informative Thing" that "**is associated with information** stored at an information store separate from the Informative Thing on a network-based system" and "**is used to determine a location** for where the information is stored and retrieve the information associated with the Informative Thing from the information store" cannot supply the necessary inventive concept. The specification notes that the idea

**UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA**

CIVIL MINUTES - GENERAL

CASE NO.: CV 16-00737 SJO (MRWx)

DATE: July 25, 2016

underlying the '933 Patent's approach—i.e. the use of a physical object that can be associated with an "identifier [that] serves as the key in a lookup table associating a physical thing with information"—"is not new." ('933 Patent col. 3 ll. 7-12.) According to the specification, the purported difference between a "Thing ID" and these known "unique identifier[s] (ID[s])," is that the known IDs "represent a **person's** identity rather than an **object's** identity[;]" thus, according to the specification, the "innovation [of the '933 Patent] is to bring the ideas of object IDs and information together as an information interface." ('933 Patent col. 3 ll. 7-24 [emphasis added].) This "innovation," however, is nothing more than taking the general concept of using identifiers with pointers to other information and "apply[ing] it on a computer," which courts have long held cannot confer patent-eligibility. See *Intellectual Ventures I*, 792 F.3d at 1370-71. The recitation of a "Thing ID" therefore does not limit the claim scope in a way that "does not solely capture the abstract idea." *Digitech*, 758 F.3d at 1350 (citing *Alice*, 134 S. Ct. 2347).

Importantly, no meaningful limitations are placed on **any** of the hardware or software components required to be used in the claimed systems, methods, and articles of manufacture. Moreover, the specification provides that "the present invention may be implemented . . . using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof." ('933 Patent col. 4:1-5.) "Given the ubiquity of computers, wholly generic computer implementation is not generally the sort of 'additional featur[e]' that provides any 'practical assurance that the process is more than a drafting effort designed to monopolize the [abstract idea] itself.'" *Alice*, 134 S. Ct. at 2358 (quoting *Mayo*, 132 S. Ct. at 1297). Additionally, the specification describes that the claimed network-based system "preferably comprises the Internet, although it could also comprise intranets, extranets, LANs, PANs, WANs, etc. A typical combination of resources may include clients that are personal computer or workstations, and servers that are personal computers, workstations, minicomputers and/or mainframes." ('933 Patent col. 3 ll. 37-43.) "The claims' invocation of [a network such as the] Internet also adds no inventive concept." *Ultramercial II*, 772 F.3d at 716; see also *buySAFE, Inc. v. Google, Inc.*, 765 F.3d 1350, 1355 (Fed. Cir. 2014) ("That a computer receives and sends the information over a network—with no further specification—is not even arguably inventive."). Narrowing the abstract idea of managing and retrieving information to a network-based system is an "attempt [] to limit the use' of the abstract idea 'to a particular technological environment,' which is insufficient to save a claim." *Ultramercial II*, 772 F.3d at 716 (citing *Alice*, 134 S. Ct. at 2358). Accordingly, none of the limitations, when viewed individually, provide an "inventive concept" sufficient to confer patent-eligibility under Section 101.

The individual claim elements, when viewed as an "ordered combination," fare no better. The claims simply recite systems, methods, and articles of manufacture that implement the abstract idea of using information stored in one place to determine the location of and retrieve information stored in a second place. See *Alice*, 134 S. Ct. at 2359 ("Viewed as a whole, petitioner's method claims simply recite the concept of intermediated settlement as performed by a generic computer. . . . [T]he claims at issue amount to 'nothing significantly more' than an instruction to apply the abstract idea of intermediated settlement using some unspecified, generic computer.").

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

CIVIL MINUTES - GENERAL

CASE NO.: CV 16-00737 SJO (MRWx)

DATE: July 25, 2016

Plaintiff fails to identify any way in which the claims "purport to improve the functioning of the computer itself" or "effect an improvement in any other technology or technical field." *Id.* Indeed, the specification itself instructs that "computationally speaking, [this method] **does not provide more power than already exists**. That is, in the networked world, any information can be accessed, but Informative Things restricts access: The user can only access information attached to a physical object." ('933 Patent col. 9 ll. 7-12 (emphasis added).) Rather, the invention purportedly "simplifies many aspects of data transfer among physically close individuals." ('933 Patent col. 11 ll. 1-4.) Plaintiff's halfhearted effort to analogize to *DDR Holdings* "for an eligibility lifeline," *IPLearn-Focus*, 2015 WL 4192092 at *5, does not persuade.

As discussed above, it is "the pre-emption concern that undergirds [] § 101 jurisprudence." *Alice*, 134 S. Ct. at 2358. Thus, as stated by another court in this district, "the question in the abstract idea context is whether there are other **ways** to use the abstract idea in the same field." *McRO, Inc. v. Sony Computer Entm't Am., LLC*, 55 F. Supp. 3d 1214, 1222 (C.D. Cal. 2014) (emphasis in original). In this case, the Court answers the question in the negative. It is difficult, if not impossible, to envision a system or method of using a separate physical device to locate information for retrieval that does not contain (1) an identifier that represents the identify of the storage device that (2) is used to retrieve associated information stored at a separate information store. That is because the '933 Patent claims the entire "starting point[] for exploiting the idea of Informative Things," which "provides the capability of attaching data to physical objects, which can be easily transported, remembered, shared, and organized." ('933 Patent col. 8 ll. 49-53.) That the independent claims do not limit how the identification is read, how the data is associated and retrieved, what types of data can be associated and retrieved, how the data must be displayed, or what non-generic or unconventional hardware or software must be used underscores the significant risk of preemption underlying the claims of the '933 Patent. Viewed in this light, the asserted claims are distinguishable from those at issue in *DDR Holdings*. *DDR Holdings*, 773 F.3d at 1259 ("It is also clear that the claims at issue do not attempt to preempt every application of the idea of increasing sales by making two web pages look the same, or of any other variant suggested by NLG. Rather, they recite a **specific way** to automate the creation of a composite web page by an 'outsource provider.'" (emphasis added)). Thus, to the extent that the "inventive concept" claimed in the '933 Patent is the ability to use generic and standard hardware and software to execute the abstract idea of using information stored in one place to determine the location of and retrieve information stored in a second place, the '933 Patent preempts the field of managing network-based information with a physical device and runs afoul of Section 101. See *TLI*, — F.3d —, 2016 WL 2865693, at *3 (finding claims invalid, and noting in its *Alice/Mayo* step 1 analysis that the asserted claims "are directed to the use of conventional or generic technology in a nascent but well-known environment, without any claim that the invention reflects an inventive solution to any problem presented by combining the two").

Viewed in this light, the claims of the '933 Patent are markedly different than those at issue in *Bascom*, which neither "preempt[ed] the use of the abstract idea of filtering content on the Internet or on generic computer components performing conventional activities." *Bascom*, — F.3d —,

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

CIVIL MINUTES - GENERAL

CASE NO.: CV 16-00737 SJO (MRWx)

DATE: July 25, 2016

2016 WL 3514158, at *8. The inventive concept in *Bascom* was found in the "non-conventional and non-generic arrangement of known, conventional pieces," which "[t]ook advantage of the ability of at least some ISPs to identify individual accounts that communicate with the ISP server, and to associate a request for Internet content with a specific individual account." *Id.* at *6. The filtering system associated individual accounts with their own, unique filtering schemes and elements, while locating the filtering system on an ISP server. *Id.* This "non-conventional and non-generic arrangement" of conventional hardware and software supplied the necessary inventive concept, and additionally required that the filtering take place on the ISP server. Here, by contrast, the claimed arrangement of elements is both generic and conventional and merely takes the abstract idea of using information stored in one place to determine the location of and retrieve information stored in a second place and says "apply it on a computer." No "inventive concept" can be found in the claim elements, whether viewed individually or as an ordered combination.

Related to the above is the question of whether a claim satisfies the "machine-or-transformation test." "While the Supreme Court has held that the machine-or-transformation test is not the sole test governing § 101 analyses, that test can provide a 'useful clue' in the second step of the *Alice* framework." *Ultramercial II*, 772 F.3d at 716 (internal citations omitted); *see also Bancorp Servs.*, 687 F.3d at 1278 (Fed. Cir. 2012) (holding that the machine-or-transformation test remains an important clue in determining whether some inventions are processes under Section 101), *cert. denied*, 134 S. Ct. 2870 (2014). "A claimed process can be patent-eligible under [Section] 101 if: '(1) it is tied to a particular machine or apparatus, or (2) it transforms a particular article into a different state or thing.'" *Ultramercial II*, 772 F.3d at 716 (quoting *Bilski*, 561 U.S. at 593).

Defendants argue that the asserted claims fail to satisfy the machine-or-transformation test, thereby confirming the '933 Patent's invalidity. (Mot. 16.) Defendants contend that the generic computer system of these claims does not give rise to a "particular machine" that would pass muster of the machine prong of the test. (Mot. 15 (citing *DealerTrack, Inc. v. Huber*, 674 F.3d 1315, 1332-34 (Fed. Cir. 2012)).) Moreover, Defendants further contend that there is no "transformation" of an article into a different state or thing, as required under the transformation prong. (Mot. 15.)

The Court agrees, concluding that the asserted claims fail the machine-or-transformation test. *Ultramercial II* itself is instructive. Here, as in *Ultramercial II*, claims 28 and 55 of the '933 Patent are "not tied to any particular novel machine, only a general purpose computer." *Ultramercial II*, 772 F.3d at 716. Instead, claims 28 and 55 generally recite a "network-based system," which is insufficient to save the patent under the machine prong of the machine-or-transformation test, particularly where the patented invention boasts the use of "personal computers or workstations" and "standard programming and/or engineering techniques" ('933 Patent cols. 3:33-4:13). *See Id.* Moreover, as in *Ultramercial II*, "[a]ny transformation from the use of computers or the transfer of content between computers is merely what computers do and does not change the analysis." *Ultramercial II*, 772 F.3d at 717. The inclusion of an "Informative Thing," which can be anything

JS-8

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

CIVIL MINUTES - GENERAL

CASE NO.: CV 16-00737 SJO (MRWx)

DATE: July 25, 2016

from a floppy disk to a memory drive to theoretically any device that "can be accessed with specialized hardware" does not pass muster under the "machine" prong.

Claim 28 likewise fails to satisfy the "transformation" prong of the machine-or-transformation test. The method as claimed refers to managing information in a network-based system by reading an ID from a data storage device and retrieving information associated with that ID. ('933 Patent col. 14 l. 62-col. 15 l. 10.) Neither reading an ID from a data storage device nor retrieving information from the network-based system meets this test because they "are not physical objects or substances, and they are not representative of physical objects or substances." *Ultramercial II*, 772 F.3d at 717. "Typically, transforming data from one form to another does not qualify as the kind of transformation regarded as an important indicator of patent eligibility." *Card Verification Solutions*, 2014 WL 4922524, at *5 (citing *CyberSource* for the proposition that "the mere manipulation or reorganization of data . . . does not satisfy the transformation prong"). And unlike in *Card Verification Solutions*, where the court, in denying the defendant's motion to dismiss, found that "the claimed invention [plausibly went] beyond manipulating, reorganizing, or collecting data by actually adding a new subset of numbers or characters to the data, thereby fundamentally altering the original confidential information," *id.* at *5, none of the limitations of claim 28 require that data be "fundamentally altered" in a manner sufficient to satisfy the transformation prong. Thus, claim 28 does not transform any article into a different state or thing. While a claim's satisfaction fo the machine-or-transformation test is not conclusive, it underscores why claim 28 recites nothing more than conventional steps of managing information over a network-based system using standard and generic hardware and software.

In conclusion, none of the asserted claims meaningfully limit the abstract idea of using information stored in one place to determine the location of and retrieve information stored in a second place, and as a result, the Court finds that each of the asserted claims is unpatentable under 35 U.S.C. section 101. Defendants have met their burden of proving that the claims of the '933 Patent are invalid as a matter of law.⁴

III. RULING

For the foregoing reasons, the Court **GRANTS** Defendants' Motion to Dismiss. This matter shall close.

IT IS SO ORDERED.

⁴ Because the Court finds that the claims of the '933 Patent are not eligible for patent protection under 35 U.S.C. section 101, the Court need not consider Defendants' arguments regarding personal jurisdiction.

EXHIBIT 7

**IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE**

ACCELERATION BAY LLC, a Delaware))	
Limited Liability Corporation,))	
)	C.A. No. 15-228-RGA
Plaintiff,))	
)	DEMAND FOR JURY TRIAL
v.))	
)	
ACTIVISION BLIZZARD, INC.,))	
a Delaware Corporation,))	
)	
Defendant.))	

**PLAINTIFF ACCELERATION BAY LLC'S
INITIAL DISCLOSURES PURSUANT TO RULE 26(a)(1)**

Plaintiff Acceleration Bay LLC (“Plaintiff”), by its undersigned counsel, hereby submits the following initial disclosures to Defendant Activision Blizzard, Inc. (“Defendant”), pursuant to Rule 26(a)(1) of the Federal Rules of Civil Procedure and Section 1 of the Court’s Rule 16 Scheduling Order dated October 29, 2015.

These disclosures are based on information now reasonably available to Plaintiff and represent a good faith effort to identify information that Plaintiff reasonably believes to be required in these disclosures. Plaintiff is continuing to investigate facts, issues, and law relevant to this action and expressly reserves the right to modify, amend, supplement and/or correct the information provided in these disclosures as information becomes available.

In making these disclosures, Plaintiff does not represent that it is identifying every fact, document, tangible thing, or witness possibly relevant to Plaintiff’s claims. These disclosures are not intended, and should not be construed, as a waiver of (i) any objection to or protection from the production, use, or admission into evidence of any document or information, that Plaintiff may be legally entitled to assert during discovery or any trial of this action; (ii) any

objection to any other discovery involving or relating to the subject matter of this disclosure; (iii) the attorney-client privilege, attorney work product doctrine, or any other applicable privilege or immunity; or (iv) the right to designate confidential information and materials for appropriate protection under any Protective Order in this action. Plaintiff also reserves the right to rely upon additional information as it becomes available through discovery or otherwise, pursuant to Fed. R. Civ. Proc. 26(e). In addition, Plaintiff reserves the right to call any witness or present any document or tangible thing at trial that is identified through further investigation or discovery.

1. Persons Having Knowledge of Facts Relevant to the Claim or Defense of Any Party

After a reasonable investigation and based on currently available information, Plaintiff identifies the individuals below as persons who may have knowledge relevant to this case. Plaintiff does not waive the attorney-client privilege, attorney work product privilege, or any other privileges in connection with the information provided herein. By providing this information, Plaintiff does not consent to Defendant and/or its counsel communicating with any of Plaintiff's current or former employees or agents. Any such individual should be contacted only through Plaintiff's counsel of record identified below.

Individual	Connection to Case and Summary of Information Known	Address/Telephone ¹
Joe Ward	CEO and Founder of Acceleration Bay LLC. Knowledge regarding Acceleration Bay's acquisition of the asserted patents and business activities.	c/o Kramer Levin Naftalis & Frankel LLP 990 Marsh Road Menlo Park, CA 94025
Joseph Agiato	Senior Vice President of Acceleration Bay. Knowledge regarding Acceleration Bay's operations and licensing activities.	c/o Kramer Levin Naftalis & Frankel LLP 990 Marsh Road Menlo Park, CA 94025

¹ The addresses and telephone numbers provided are those last known to Plaintiff. For individuals located at corporate entities, the corporate headquarters (or other corporate offices) have been provided in cases where the particular business address of the individual is presently unknown.

Individual	Connection to Case and Summary of Information Known	Address/Telephone ¹
Drew Anderson	Vice President of Acceleration Bay. Knowledge regarding Acceleration Bay's research and development activities.	c/o Kramer Levin Naftalis & Frankel LLP 990 Marsh Road Menlo Park, CA 94025
Fred Holt	Named inventor. Knowledge regarding the technology contained in one or more of the asserted patents and/or related patents and applications.	c/o Kramer Levin Naftalis & Frankel LLP 990 Marsh Road Menlo Park, CA 94025
Virgil Bourassa	Named inventor. Knowledge regarding the technology contained in one or more of the asserted patents and/or related patents and applications.	c/o Kramer Levin Naftalis & Frankel LLP 990 Marsh Road Menlo Park, CA 94025
Linda Magnotti	Executive at Panthesis, Inc. Knowledge regarding Panthesis, Inc.	c/o Kramer Levin Naftalis & Frankel LLP 990 Marsh Road Menlo Park, CA 94025
Maurice Pirio	Prosecuting attorney. Knowledge regarding prosecution of one or more of the asserted patents and/or related patents and applications.	Perkins Coie LLP P.O. Box 1247 Seattle, WA 98111
Robert Kotick	President and CEO of Activision Blizzard. Knowledge regarding the accused products, infringement and damages.	Activision Blizzard, Inc. 3100 Ocean Park Boulevard Santa Monica, CA 90405
Dennis Durkin	CFO of Activision Blizzard. Knowledge regarding the accused products, infringement and damages.	Activision Blizzard, Inc. 3100 Ocean Park Boulevard Santa Monica, CA 90405
Eric Hirshberg	President and CEO of Activision Publishing. Knowledge regarding the accused products, infringement and damages.	Activision Blizzard, Inc. 3100 Ocean Park Boulevard Santa Monica, CA 90405
Brian Hodous	Chief Customer Officer of Activision Blizzard. Knowledge regarding the accused products, infringement and damages.	Activision Blizzard, Inc. 3100 Ocean Park Boulevard Santa Monica, CA 90405
Michael Morhaime	CEO of Blizzard Entertainment. Knowledge regarding the accused products, infringement and damages.	Activision Blizzard, Inc. 3100 Ocean Park Boulevard Santa Monica, CA 90405
Humam Sakhini	Chief Strategy and Talent Officer of Activision Blizzard. Knowledge regarding the accused products, infringement and damages.	Activision Blizzard, Inc. 3100 Ocean Park Boulevard Santa Monica, CA 90405

Individual	Connection to Case and Summary of Information Known	Address/Telephone¹
Thomas Tippel	COO of Activision Blizzard. Knowledge regarding the accused products, infringement and damages.	Activision Blizzard, Inc. 3100 Ocean Park Boulevard Santa Monica, CA 90405
Tim Ellis	EVP Chief Marketing Officer at Activision. Knowledge regarding the accused products, infringement and damages.	Activision Blizzard, Inc. 3100 Ocean Park Boulevard Santa Monica, CA 90405
Other employees and officers of Defendant	Knowledge regarding the operation, marketing and sales of the accused products, services, and networks; Defendant's licensing practices. The value of the claimed inventions to Defendant.	TBD
Plaintiff's expert witnesses	Plaintiff's expert witnesses will be disclosed and will provide expert reports in connection with this litigation at the appropriate time.	TBD
Any expert witness that Defendant retains	TBD	TBD
Any person and/or entity deposed in this case	TBD	TBD
Any person or entity identified in Defendant's Initial Disclosures and/or disclosed through discovery response or documents produced in this matter	TBD	TBD
As of yet unknown representatives of third parties.	TBD	TBD

Plaintiff will supplement this section as the names and contact information for additional parties with relevant information become available. Plaintiff reserves the right to seek discovery from and relating to such additional parties who become known. Plaintiff also reserves the right to rely on expert testimony regarding infringement of U.S. Patent Nos. 6,701,344; 6,714,966;

6,732,147; 6,829,634; 6,910,069; and 6,920,497 (collectively, the “Patents-in-Suit”), the validity of the Patents-in-Suit, and damages.

In making these disclosures, Plaintiff does not waive its right to object, pursuant to the applicable Federal and Local Rules, to discovery of information from any of the individuals listed above.

2. Documents Relevant to the Claims or Defenses of Any Party

Plaintiff sets forth the following description of categories of documents in Plaintiff’s possession, custody, or control that it may use to support its claims or defenses. Because Defendant has not identified all of the bases for its defenses and counterclaims, and because Plaintiff’s review of the allegations in this matter is ongoing, Plaintiff reserves the right to identify additional documents as discovery proceeds:

- The Patents-in-Suit
- File histories of the Patents-in-Suit
- Documents related to infringement of the Patents-in-Suit, including information posted on Defendant’s websites or in product manuals or technical documents.
- Relevant emails between Acceleration Bay’s employees or officers.
- Acceleration Bay’s public documents, including information posted on its website.
- Certain licenses related to the Patents-in-Suit or this action.

3. Computation of Damages

Plaintiff seeks all damages to which it is entitled under the Patent Laws, including 35 U.S.C. § 284, arising from Defendant’s infringement. Plaintiff seeks damages in an amount adequate to compensate for the infringement, which includes, but is not limited to, a reasonable royalty for the use of the invention, together with interest and costs fixed by the Court. Plaintiff additionally seeks an accounting of all infringing sales and revenues. Plaintiff also seeks an

award of attorney's fees and expenses associated with the present action under 35 U.S. C. § 285. Such costs, fees, and expenses cannot be computed at the present time and depend on a variety of factors such as the length and intensity of the litigation and the positions that Defendant takes.

Since much of the information necessary to make damages calculations is in the possession of Defendant, Plaintiff reserves its right to identify additional documents as this matter proceeds. Moreover, Defendant's patent infringement is ongoing and the amount of damages to which Plaintiff is entitled continues to grow.

4. Insurance Agreements

As presently known, there is no indemnity or insuring agreement under which any person or entity carrying on an insurance business may be liable to satisfy part or all of a judgment entered in this action or to indemnify or reimburse for payments made to satisfy the judgment.

POTTER ANDERSON & CORROON LLP

OF COUNSEL:

Paul J. Andre
Lisa Kobialka
James R. Hannah
KRAMER LEVIN NAFTALIS &
FRANKEL LLP
990 Marsh Road
Menlo Park, CA 94025
(650) 752-1700

Dated: November 2, 2015
1208110

By: /s/ Philip A. Rovner
Philip A. Rovner (#3215)
Jonathan A. Choa (#5319)
Hercules Plaza
P.O. Box 951
Wilmington, DE 19899
(302) 984-6000
provner@potteranderson.com
jchoa@potteranderson.com

Attorneys for Plaintiff Acceleration Bay LLC

CERTIFICATE OF SERVICE

I, Philip A. Rovner, hereby certify that, prior to 6 p.m. on November 2, 2015, 2015, the within document was served on the following counsel as indicated:

BY E-MAIL

Jack B. Blumenfeld, Esq.
Stephen J. Kraftschik, Esq.
Morris, Nichols, Arsht & Tunnell LLP
1201 N. Market Street
P.O. Box 1347
Wilmington, DE 19899
jblumenfeld@mnat.com
skraftschik@mnat.com

Attorneys for Defendants

Michael A. Tomasulo, Esq.
David P. Enzminger, Esq.
David K. Lin, Esq.
Gino Cheng, Esq.
Winston & Strawn LLP
333 S. Grand Avenue
Los Angeles, CA 90071
mtomasulo@winston.com
denzminger@winston.com
dlin@winston.com
gcheng@winston.com

Co-counsel for Defendants

Daniel K. Webb, Esq.
Winston & Strawn LLP
35 W. Wacker Drive
Chicago, IL 60601
dwebb@winston.com

Co-counsel for Defendants

/s/ Philip A. Rovner
Philip A. Rovner (#3215)
Potter Anderson & Corroon LLP
Hercules Plaza
P. O. Box 951
Wilmington, DE 19899
(302) 984-6000
provner@potteranderson.com

EXHIBIT 8

**THIS EXHIBIT HAS BEEN
REDACTED IN ITS ENTIRETY**

EXHIBIT 9

**THIS EXHIBIT HAS BEEN
REDACTED IN ITS ENTIRETY**

EXHIBIT 10

**THIS EXHIBIT HAS BEEN
REDACTED IN ITS ENTIRETY**

EXHIBIT 11

**THIS EXHIBIT HAS BEEN
REDACTED IN ITS ENTIRETY**

EXHIBIT 12

**THIS EXHIBIT HAS BEEN
REDACTED IN ITS ENTIRETY**

EXHIBIT 13

**THIS EXHIBIT HAS BEEN
REDACTED IN ITS ENTIRETY**

EXHIBIT 14

Network Serialization and Routing in World of Warcraft



Joe Rumsey
jrumsey@blizzard.com
Twitter: @joerumz

GDC
13



What is JAM?

Joe's
Automated
Messages

GDC
13



The Problem

Game servers need to communicate with each other

GDC
13

BLIZZARD
ENTERTAINMENT

Manual serialization is error-prone

```
void Serialize(stream &msg)
{
    vector<int> values;
    // ...Fill in some values...
    msg << values.size();
    for(int i = values.size(); --i;)
    {
        msg << values[i];
    }
}

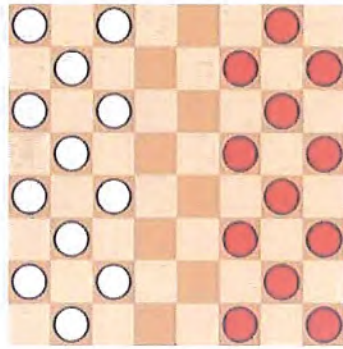
void Deserialize(stream &msg)
{
    vector<int> values;
    int size;
    msg >> size;
    values.resize(size);
    for(int i = size; i--;)
    {
        msg >> values[i];
    }
}
```

Manual serialization is error-prone

```
void Serialize(stream &msg)
{
    vector<int> values;
    // ... Fill in some values...
    msg << values.size();
    for(int i = values.size(); --i;)
    {
        msg << values[i];
    }
}

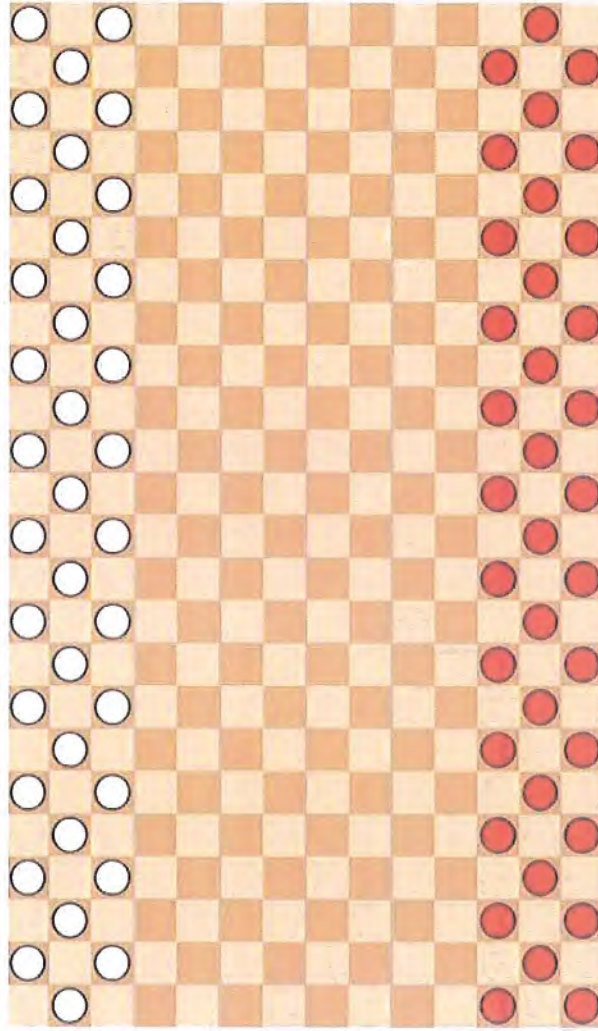
void Deserialize(stream &msg)
{
    vector<int> values;
    int size;
    msg >> size;
    values.resize(size);
    for(int i = size; i--;)
    {
        msg >> values[i];
    }
}
```

Manual serialization doesn't scale



Manual serialization doesn't scale

World Of Checkers

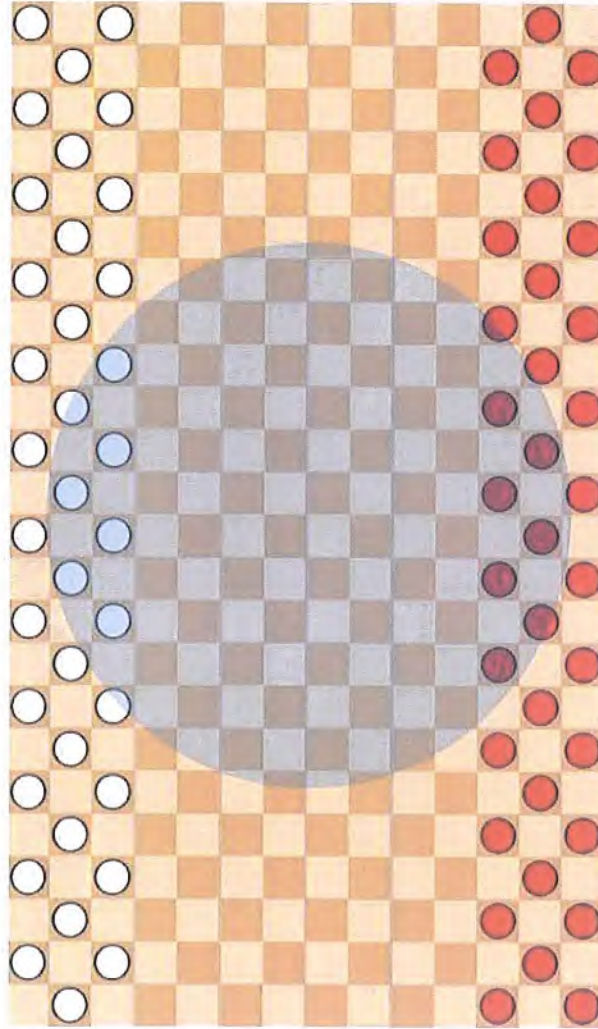


GOC¹³



Manual serialization doesn't scale

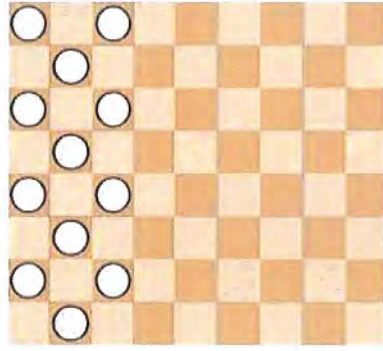
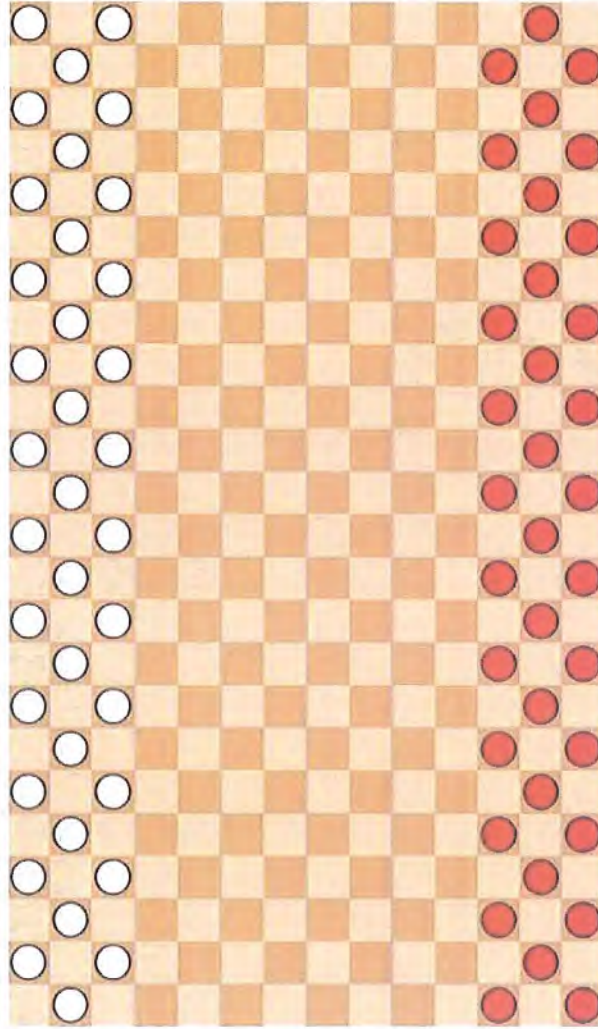
World Of Checkers



GDC¹³

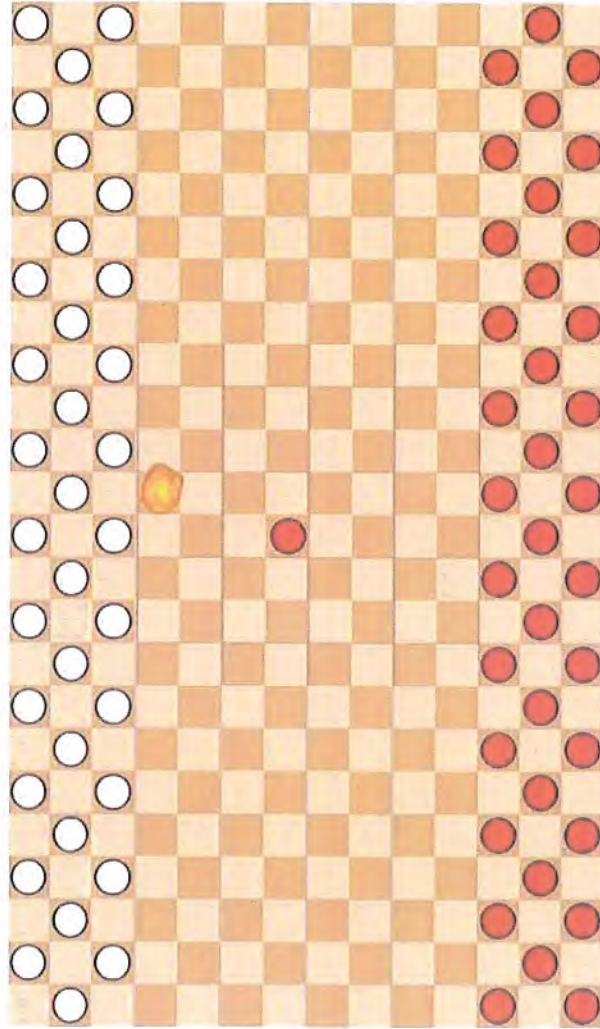
WIZARD ENTERTAINMENT

Manual serialization doesn't scale World Of Checkers



Manual serialization doesn't scale

World Of Checkers

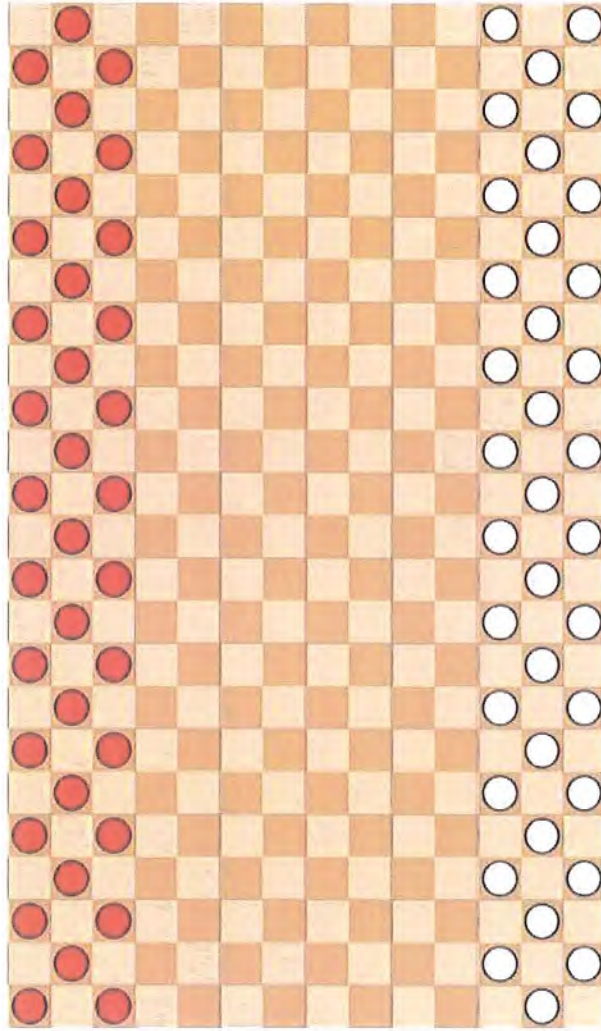


GDC¹³

WIZARD
ENTERTAINMENT

Manual serialization doesn't scale

World Of Checkers



GOC
13



Goals

GOC
13

BIZAM
ENTERTAINMENT

Goals

- DRY - Don't Repeat Yourself

Goals

- DRY - Don't Repeat Yourself
- Eliminate boilerplate to reduce bugs

Goals

- DRY - Don't Repeat Yourself
- Eliminate boilerplate to reduce bugs
- No more hand-coded serialize/deserialize

Goals

- DRY - Don't Repeat Yourself
- Eliminate boilerplate to reduce bugs
- No more hand-coded serialize/deserialize
- Spend more time on the game, not the protocol

Goals

- DRY - Don't Repeat Yourself
- Eliminate boilerplate to reduce bugs
- No more hand-coded serialize/deserialize
- Spend more time on the game, not the protocol
- Build a helpful robot that writes our code for us

Goal: Human readable code

```
struct CheckerCaptured {  
    CheckerID id;  
    CheckerID capturedBy;  
    u8 jumpType;  
};  
  
void Capture(CheckerID id, CheckerID by, JUMP_TYPE jumpType)  
{  
    CheckerCaptured msg;  
    msg.id = id;  
    msg.capturedBy = by;  
    msg.jumpType = jumpType;  
    Send(&msg);  
}
```



Goal: Human readable code

```
struct CheckerCaptured {  
    CheckerID id;  
    CheckerID capturedBy;  
    u8 jumpType;  
};
```

```
void Capture(CheckerID id, CheckerID by, JUMP_TYPE jumpType)  
{  
    CheckerCaptured msg;  
    msg.id = id;  
    msg.capturedBy = by;  
    msg.jumpType = jumpType;  
    Send(&msg);  
}
```



Goal: Human readable code

```
struct CheckerCaptured {
    CheckerID id;
    CheckerID capturedBy;
    u8 jumpType;
};

void Capture(CheckerID id, CheckerID by, JUMP_TYPE jumpType)
{
    CheckerCaptured msg;
    msg.id = id;
    msg.capturedBy = by;
    msg.jumpType = jumpType;
    Send(&msg);
}
```

Goal: Human readable code

```
struct CheckerCaptured {
    CheckerID id;
    CheckerID capturedBy;
    u8 jumpType;
};

void Capture(CheckerID id, CheckerID by, JUMP_TYPE jumpType)
{
    CheckerCaptured msg;
    msg.id = id;
    msg.capturedBy = by;
    msg.jumpType = jumpType;
    Send(&msg);
}
```

Goal: Human readable code

```
struct CheckerCaptured {  
    CheckerID id;  
    CheckerID capturedBy;  
    u8 jumpType;  
};  
  
void Capture(CheckerID id, CheckerID by, JUMP_TYPE jumpType)  
{  
    CheckerCaptured msg;  
    msg.id = id;  
    msg.capturedBy = by;  
    msg.jumpType = jumpType;  
    Send(&msg);  
}
```


Goal: Human readable code

```
struct CheckerCaptured {
    CheckerID id;
    CheckerID capturedBy;
    u8 jumpType;
};

void Capture(CheckerID id, CheckerID by, JUMP_TYPE jumpType)
{
    CheckerCaptured msg;
    msg.id = id;
    msg.capturedBy = by;
    msg.jumpType = jumpType;
    Send(&msg);
}
```

Implementation Details

GDC
13



Development Cycle

- Describe the protocol
- Generate serialization and dispatch
- Send messages
- Receive messages
- Configure routing info

I-to-I mapping of .jam messages to C++ classes

```
// From Checkers.jam  
message CheckerCapturedCredit {  
    CheckerID capturedCheckerID;  
    CheckerID capturedBy;  
    u8 jumpType;  
};
```


I-to-I mapping of .jam messages to C++ classes

```

// 100% Generated code in JamCheckers.cpp
class CheckerCaptureCredit : public JamMessage {
public:
    // Message decoders
    BOOL Get(BinaryDecoder &decoder);
    BOOL Get(JSONDecoder &decoder);
    // Message encoders
    BOOL Put(BinaryEncoder &encoder) const;
    BOOL Put(JSONEncoder &encoder) const;
    /**** DATA START ****/
    CheckerID capturedCheckerID;
    CheckerID capturedBy;
    u8 jumpType;
    /**** DATA STOP ****/
    // Lots more stuff...
};

// From Checkers.jam
message CheckerCaptureCredit {
    CheckerID capturedCheckerID;
    CheckerID capturedBy;
    u8 jumpType;
};

```

Development Cycle

- Describe the protocol
- **Generate serialization and dispatch**
- Send messages
- Receive messages
- Configure routing info

Auto-generated serialization code

```
//NOTICE: This is generated code. DO NOT EDIT!  
BOOL CheckerCaptureCredit::Put(BinaryEncoder &_encoder) const  
{  
    _encoder.BeginMessage(CODE, NAME);  
    _encoder.Put("capturedCheckerID", capturedCheckerID);  
    _encoder.Put("capturedBy", capturedBy);  
    _encoder.Put("jumpType", jumpType);  
    _encoder.EndMessage(CODE, NAME);  
    return TRUE;  
}
```

Flex and Bison make writing parsers easy



Other tools

- ANTLR
- GOLD
- PLY (Python Lex & Yacc)
- Boost.Spirit

Flex & Bison - parser generators

JAM File syntax is described to Bison

```

jamstructs      : jamstructs jamstruct
                 | jamstruct
                 ;

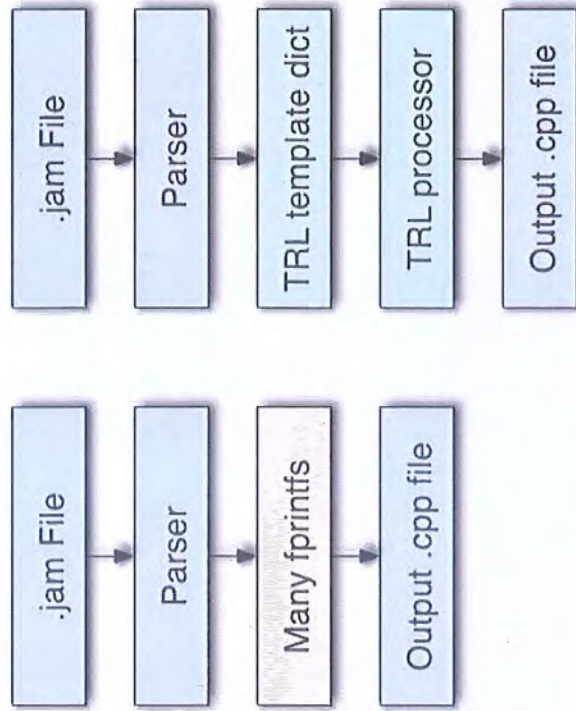
jamstruct       : structtype TIDENTIFIER messagename '{' { jam_start_s
                 | structtype TIDENTIFIER messagename '{' ' '; ' { jar
                 | jamcomment { /*printf("Comment\n");*/ }
                 ;

structtype      : TMESSAGE { $$ .type = GMESSAGE; $$ .qualifiers = 0; }
                 | TOBJECTMESSAGE { $$ .type = GMESSAGE; $$ .qualifiers =
                 | TSTRUCT { $$ .type = GSTRUCT; $$ .qualifiers = 0; }
                 ;

```




From .jam to .cpp



2004

2013

TRL Turns .jam into C++

```
{@ define OutputMessage(msg, encoders, decoders) @}
//
// NOTICE: This is generated code. DO NOT EDIT!
//
class {{ msg.structName }} : public JamMessage {
public:
    static u32 CRC;
    static u16 CODE;
    static cchar *NAME;

    // No argument constructor:
    {{ msg.structName }}() {

        {@ foreach f in msg.fields @}
        {@ if f.hasDefault @}
            {{ f.name }} = {{ f.defValue }};
        {@ end if @}
        {@ end foreach @}

    }
}
```

TRL to generate a message
class definition

TRL Turns .jam into C++

```
{@ define OutputMessage(msg, encoders, decoders) @}
//
// NOTICE: This is generated code. DO NOT EDIT!
//
class {{ msg.structName }} : public JamMessage {
public:
    static u32 CRC;
    static u16 CODE;
    static cchar *NAME;

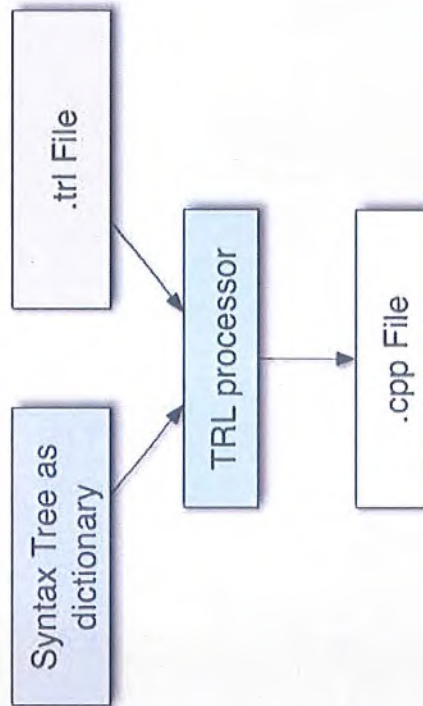
    // No argument constructor:
    {{ msg.structName }}() {
        {@ foreach f in msg.fields @}
        {@ if f.hasDefault @}
            {{ f.name }} = {{ f.defaultValue }};
        {@ end if @}
        {@ end foreach @}
    }
}
```

TRL to generate a message
class definition

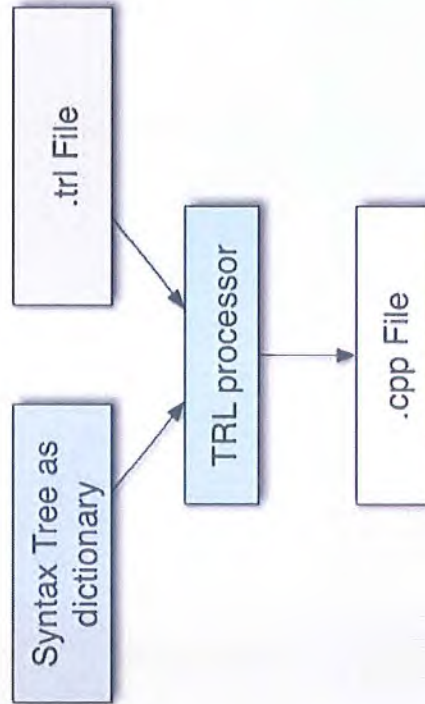
See Also

- CTemplate
- ngTemplate
- Django (HTML focused)
- Jinja (Python)

Fill out a dictionary and feed it to TRL

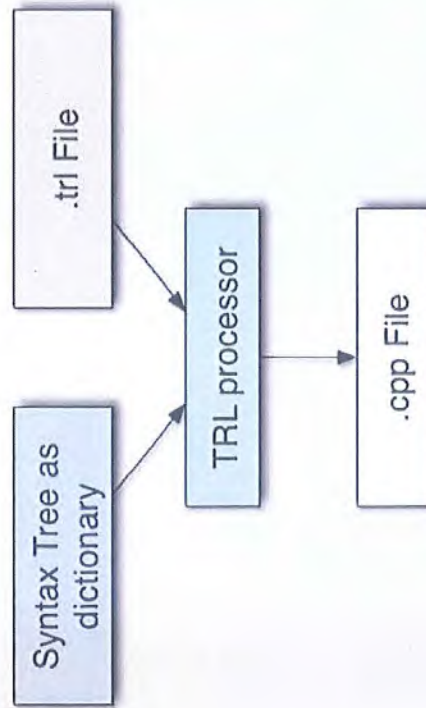


Fill out a dictionary and feed it to TRL



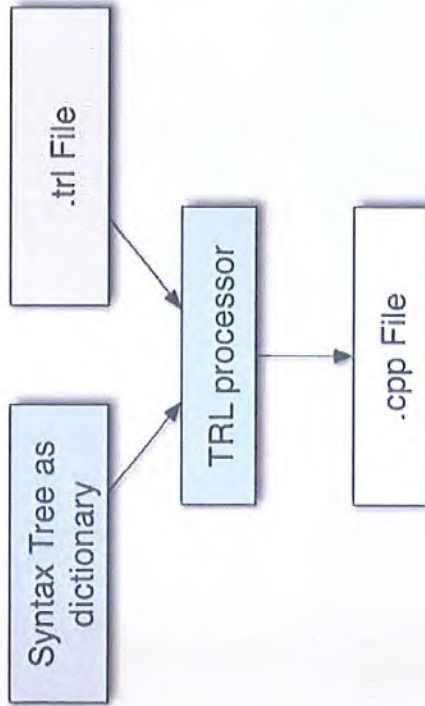
```
{@ foreach f in msg.fields @}  
  {@ if f.hasDefault @  
    {{ f.name }} = {{ f.defaultValue }};  
  @ end if @}  
{@ end foreach @}
```

Fill out a dictionary and feed it to TRL



```
{@ foreach f in msg.fields @  
  {@ if f.hasDefault @  
    {{ f.name }} = {{ f.defaultValue }};  
    end if @  
  end foreach @}
```


Fill out a dictionary and feed it to TRL



```
{@ foreach f in msg.fields @  
  {@ if f.hasDefault @  
    {{ f.name }} = {{ f.defaultValue }};  
  {@ end if @  
  {@ end foreach @}
```

Global Feature addition using TRL

```

{@ end foreach @}
// Virtual encode/decode dispatchers
BOOL {{m.structName}}::Put(teDataChain *pPayl
switch(protocolType) {
default:
return FALSE;
case JAM_PROTOCOL_BINARY_LITERAL:
{
teRawBinaryEncoder encoder(pPayl
return Put(encoder);
}
case JAM_PROTOCOL_TEXT_JSON:
{
teJSONEncoder encoder(pPayload);
return Put(encoder);
}
}

```

```

177 // Virtual encode/decode dispatchers
178 BOOL {{m.structName}}::Put(teDataChain *pPayl
179 switch(protocolType) {
180 default:
181 return FALSE;
182 case JAM_PROTOCOL_BINARY_COMPRESSED:
183 case JAM_PROTOCOL_BINARY_LITERAL:
184 {
185 teRawBinaryEncoder encoder(pPayl
186 return Put(encoder);
187 }
188 case JAM_PROTOCOL_TEXT_JSON_COMPRESSED:
189 case JAM_PROTOCOL_TEXT_JSON:
190 {
191 teJSONEncoder encoder(pPayload);

```



Development Cycle

- Describe the protocol
- Generate serialization and dispatch
- Send messages
- Receive messages
- Configure routing info

Create a message, fill in data, call send

```
void Checker::OnCaptured(CheckerID capturedBy, JUMP_TYPE how)
{
    CheckerCapturedCredit msg;
    msg.capturedCheckerID = GetID();
    msg.capturedBy = capturedBy;
    msg.jumpType = how;
    JamID destination = GetRouter()->GetCreditManagerID();
    GetRouter()->Send(destination, &msg);
}
```


Create a message, fill in data, call send

```
void Checker::OnCaptured(CheckerID capturedBy, JUMP_TYPE how)
{
    CheckerCapturedCredit msg;
    msg.capturedCheckerID = GetID();
    msg.capturedBy = capturedBy;
    msg.jumpType = how;
    JamID destination = GetRouter()->GetCreditManagerID();
    GetRouter()->Send(destination, &msg);
}
```


Create a message, fill in data, call send

```
void Checker::OnCaptured(CheckerID capturedBy, JUMP_TYPE how)
{
    CheckerCapturedCredit msg;
    msg.capturedCheckerID = GetID();
    msg.capturedBy = capturedBy;
    msg.jumpType = how;
    JamID destination = GetRouter()->GetCreditManagerID();
    GetRouter()->Send(destination, &msg);
}
```

Create a message, fill in data, call send

```
void Checker::OnCaptured(CheckerID capturedBy, JUMP_TYPE how)
{
    CheckerCapturedCredit msg;
    msg.capturedCheckerID = GetID();
    msg.capturedBy = capturedBy;
    msg.jumpType = how;
    JamID destination = GetRouter()->GetCreditManagerID();
    GetRouter()->Send(destination, &msg);
}
```



Create a message, fill in data, call send

```
void Checker::OnCaptured(CheckerID capturedBy, JUMP_TYPE how)
{
    CheckerCapturedCredit msg;
    msg.capturedCheckerID = GetID();
    msg.capturedBy = capturedBy;
    msg.jumpType = how;
    JamID destination = GetRouter()->GetCreditManagerID();
    GetRouter()->Send(destination, &msg);
}
```

Create a message, fill in data, call send

```
void Checker::OnCaptured(CheckerID capturedBy, JUMP_TYPE how)
{
    CheckerCapturedCredit msg;
    msg.capturedCheckerID = GetID();
    msg.capturedBy = capturedBy;
    msg.jumpType = how;
    JamID destination = GetRouter()->GetCreditManagerID();
    GetRouter()->Send(destination, &msg);
}
```



Structs and arrays in messages

```
message GroupUpdate  
{  
    GroupID group;  
    array<.CheckerID> checkers;  
};
```

```
/**/ DATA START ***/  
GroupID group;  
vector<CheckerID> checkers;  
/**/ DATA STOP ***/
```

```
void GroupService::SendUpdate(GroupID id)  
{  
    GroupUpdate msg;  
    msg.group = id;  
    msg.checkers.resize(MAX_GROUP_SIZE);  
    // ...  
}
```


Structs and arrays in messages

```
message GroupUpdate  
{  
    GroupID group;  
    array<.CheckerID> checkers;  
};
```

```
/**/ DATA START /**/  
GroupID group;  
vector<CheckerID> checkers;  
/**/ DATA STOP /**/
```

```
void GroupService::SendUpdate(GroupID id)  
{  
    GroupUpdate msg;  
    msg.group = id;  
    msg.checkers.resize(MAX_GROUP_SIZE);  
    // ...  
}
```

Structs and arrays in messages

```
message GroupUpdate  
{  
    GroupID group;  
    array<.CheckerID> checkers;  
};
```

```
/**/ DATA START ***/  
GroupID group;  
vector<CheckerID> checkers;  
/**/ DATA STOP ***/
```

```
void GroupService::SendUpdate(GroupID id)  
{  
    GroupUpdate msg;  
    msg.group = id;  
    msg.checkers.resize(MAX_GROUP_SIZE);  
    // ...  
}
```


Structs and arrays in messages

```
message GroupUpdate  
{  
    GroupID group;  
    array<.CheckerID> checkers;  
};
```

```
/**/ DATA START /**/  
GroupID group;  
vector<CheckerID> checkers;  
/**/ DATA STOP /**/
```

```
void GroupService::SendUpdate(GroupID id)  
{  
    GroupUpdate msg;  
    msg.group = id;  
    msg.checkers.resize(MAX_GROUP_SIZE);  
    // ...  
}
```



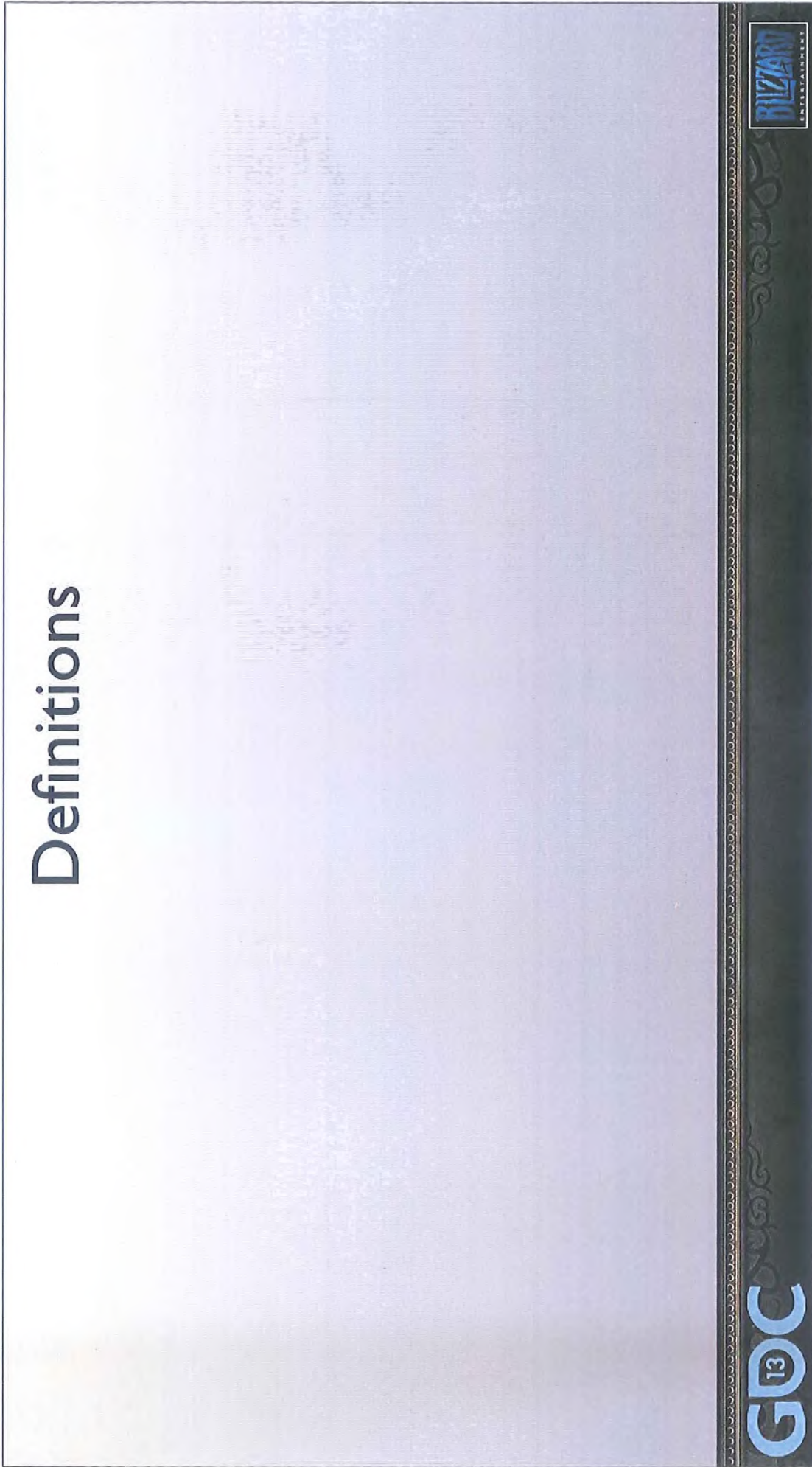
Structs and arrays in messages

```
message GroupUpdate  
{  
    GroupID group;  
    array<.CheckerID> checkers;  
};
```

```
/**/ DATA START /**/  
GroupID group;  
vector<CheckerID> checkers;  
/**/ DATA STOP /**/
```

```
void GroupService::SendUpdate(GroupID id)  
{  
    GroupUpdate msg;  
    msg.group = id;  
    msg.checkers.resize(MAX_GROUP_SIZE);  
    // ...  
}
```


Definitions



GOC
13

HIZARD
ENTERTAINMENT

Definitions

- Message - serialized structure defined in a .jam file

Definitions

- Message - serialized structure defined in a .jam file
- Protocol - a collection of messages

Definitions

- Message - serialized structure defined in a .jam file
- Protocol - a collection of messages
- Service - a module of code that implements message handlers for one or more protocols

Definitions

- Message - serialized structure defined in a .jam file
- Protocol - a collection of messages
- Service - a module of code that implements message handlers
for one or more protocols
- Program - can be composed of multiple services

Message Destinations

GDC
13

RIZZANI
ENTERTAINMENT

AB-AB 002880

Message Destinations

```
void MatchService::CreateBoard(u64 width, u64 height) {  
    BoardID = GenerateBoard();  
    // Send to a known, connected, service  
    m_pServer->Send(m_boardServerID, &msg);  
}
```

Message Destinations

```
void MatchService::CreateBoard(u64 width, u64 height) {
    BoardID = GenerateBoard();
    // Send to a known, connected, service
    m_pServer->Send(m_boardServerID, &msg);
}

void MatchService::GameOver(u32 gameId, u64 winnerID) {
    msg.gameID = gameId;
    msg.winner = winnerID();
    // Send to a service type, non-specified ID
    m_pServer->Send(JAM_SERVER_STATS_TRACKER, &msg);
}
```


Message Destinations

```
void MatchService::CreateBoard(u64 width, u64 height) {
    BoardID = GenerateBoard();
    // Send to a known, connected, service
    m_pServer->Send(m_boardServerID, &msg);
}

void MatchService::GameOver(u32 gameId, u64 winnerID) {
    msg.gameID = gameId;
    msg.winner = winnerID();
    // Send to a service type, non-specified ID
    m_pServer->Broadcast(JAM_SERVER_STATS_TRACKER, &msg);
}
```


Message Destinations

```
void MatchService::CreateBoard(u64 width, u64 height) {
    BoardID = GenerateBoard();
    // Send to a known, connected, service
    m_pServer->Send(m_boardServerID, &msg);
}

void MatchService::GameOver(u32 gameId, u64 winnerID) {
    msg.gameID = gameId;
    msg.winner = winnerID();
    // Send to a service type, non-specified ID
    m_pServer->Broadcast(JAM_SERVER_STATS_TRACKER, &msg);
}

void Checker::HealChecker(CheckerID toHeal, u32 amount) {
    CheckerHeal msg;
    msg.healedBy = GetID();
    msg.amount = amount;
    // Send a message to a specific object
    m_pServer->Send(toHeal, &msg);
}
```

Message routing by type

```
MatchmakerAddPlayer addMsg;  
addMsg.player = GetPlayerID();  
addMsg.rank = GetRank();  
  
// No JamID needed, send to any Matchmaker  
// May be queued until a Matchmaker is available  
m_pService->Send(JAM_SERVER_MATCHMAKER, &addMsg);
```


Send a message and expect a response

```
MatchmakerAddPlayer addMsg;  
addMsg.player = GetPlayerID();  
addMsg.level = GetLevel();  
  
// Send to any Matchmaker, PlayerAddedHandler  
// will be called with response when complete  
m_pService->SendRegistered<PlayerAdded>(  
    JAM_SERVER_MATCHMAKER, &addMsg  
);
```

Send a message and expect a response

```
MatchmakerAddPlayer addMsg;  
addMsg.player = GetPlayerID();  
addMsg.level = GetLevel();  
  
// Send to any Matchmaker, PlayerAddedHandler  
// will be called with response when complete  
m_pService->SendRegistered<PlayerAdded>(  
    JAM_SERVER_MATCHMAKER, &addMsg  
);
```


Send a message to an object

```
void CheckerGroup::ChangeBoards(u32 newBoard)
{
    CheckerChangeBoard msg;
    msg.boardID = newBoard;
    for(int i = 0; i < m_checkers.size(); i++) {
        m_pServer->Send(m_checkers[i]->GetID(), &msg);
    }
}
```

Each object is owned by one server

```
class Checker {  
    //...  
    CheckerID m_id;  
    JamID m_serverID;  
  
    JamID GetServer() {  
        return m_serverID;  
    }  
  
    CheckerID GetID() {  
        return m_id;  
    }  
    //...  
};
```

Each object is owned by one server

```
class Checker {  
    //...  
    CheckerID m_id;  
    JamID m_serverID;  
  
    JamID GetServer() {  
        return m_serverID;  
    }  
  
    CheckerID GetID() {  
        return m_id;  
    }  
    //...  
};
```



Each object is owned by one server

```
class Checker {  
    //...  
    CheckerID m_id;  
    JamID m_serverID;  
  
    JamID GetServer() {  
        return m_serverID;  
    }  
  
    CheckerID GetID() {  
        return m_id;  
    }  
    //...  
};
```


How messages get routed

```
void BoardServer::Send(Checker *pChecker, JamMessage *pMessage)
{
    m_pJamServer->Send(pChecker->GetServer(),
                      pChecker->GetID(),
                      pMessage);
}
```

Development Cycle

- Describe the protocol
- Generate serialization and dispatch
- Send messages
- Receive messages
- Configure routing info

On receipt, look up and dispatch

```
// static callback registered with JAM by protocol ID
// called for each incoming message
void BoardServer::CheckerDispatch(JamLink &link, JamMessage *pMessage)
{
    CheckerID destID = pMessage->GetDestination();
    Checker *pChecker = GetCheckerObject(destID);
    pChecker->QueueMessage(pMessage);
    switch (pMessage->GetProtocolCRC()) {
        case JAMCheckerProtocol_CRC:
            JamCheckerProtocol::Dispatch<Checker>(pMessage, pChecker);
    }
}
```

On receipt, look up and dispatch

```
// static callback registered with JAM by protocol ID
// called for each incoming message
void BoardServer::CheckerDispatch(JamLink &link, JamMessage *pMessage)
{
    CheckerID destID = pMessage->GetDestination();
    Checker *pChecker = GetCheckerObject(destID);
    pChecker->QueueMessage(pMessage);
    switch (pMessage->GetProtocolCRC()) {
        case JAMCheckerProtocol_CRC:
            JamCheckerProtocol::Dispatch<Checker>(pMessage, pChecker);
    }
}
```


On receipt, look up and dispatch

```
// static callback registered with JAM by protocol ID
// called for each incoming message
void BoardServer::CheckerDispatch(JamLink &link, JamMessage *pMessage)
{
    CheckerID destID = pMessage->GetDestination();
    Checker *pChecker = GetCheckerObject(destID);
    pChecker->QueueMessage(pMessage);
    switch (pMessage->GetProtocolCRC()) {
        case JAMCheckerProtocol_CRC:
            JamCheckerProtocol::Dispatch<Checker>(pMessage, pChecker);
    }
}
```

JamLink

```
void BoardServer::CheckerDispatch(JamLink &Link,  
JamMessage *pMessage)
```

```
{
```

GDC
13

RIZARD
ENTERTAINMENT

Generated Dispatch methods

```
//NOTICE: This is generated code. DO NOT EDIT!  
template<typename HANDLER_T>  
static JAM_RESULT Dispatch(JamMessage *pMessage,   
                           HANDLER_T *pHandler) {  
    switch (pMessage->GetCode()) {  
    case JAM_MSG_CheckerHeal:  
        result = pHandler->CheckerHealHandler(Link,  
        (CheckerHeal *)pMessage);  
        break;  
    // cases for rest of protocol's messages...
```

Generated Dispatch methods

```
//NOTICE: This is generated code. DO NOT EDIT!  
template<typename HANDLER_T>  
static JAM_RESULT Dispatch(JamMessage *pMessage,   
                           HANDLER_T *pHandler) {  
    switch (pMessage->GetCode()) {  
    case JAM_MSG_CheckerHeal:  
        result = pHandler->CheckerHealHandler(Link,  
        (CheckerHeal *)pMessage);  
        break;  
    // cases for rest of protocol's messages...
```


Generated Dispatch methods

```
//NOTICE: This is generated code. DO NOT EDIT!  
template<typename HANDLER_T>  
static JAM_RESULT Dispatch(JamMessage *pMessage,   
                           HANDLER_T *pHandler) {  
    switch (pMessage->GetCode()) {  
    case JAM_MSG_CheckerHeal:  
        result = pHandler->CheckerHealHandler(Link,  
        (CheckerHeal *)pMessage);  
        break;  
    // cases for rest of protocol's messages...
```

Generated message handler prototypes

```
// A message handler prototype is auto-generated for each message
// in the protocol. #include these declarations in the middle
// of your hand constructed class.
JAM_RESULT CheckerHealHandler(JamLink &Link, CheckerHeal *msg);
JAM_RESULT CheckerDamageHandler(JamLink &Link, CheckerDamage *msg);
JAM_RESULT CheckerPowerupHandler(JamLink &Link, CheckerPowerup *msg);
JAM_RESULT CheckerKingHandler(JamLink &Link, CheckerKing *msg);
```

#include this in the middle of a class

Message handler methods

```
JAM_RESULT Checker::CheckerHealHandler(CheckerHeal *pMessage)
{
    m_health += pMessage->amount;
    LOG("Checker %d was healed for %d by checker %d",
        GetID(), pMessage->amount, pMessage->healedBy);
    return JAM_OK;
}
```

Send and Receive

```
void Checker::HealChecker(CheckerID toHeal, u32 amount) {
    CheckerHeal msg;
    msg.healedBy = GetID();
    msg.amount = amount;
    // Send a message to a specific object
    m_pServer->Send(toHeal, &msg);
}

JAM_RESULT Checker::CheckerHandler(CheckerHeal *pMessage)
{
    m_health += pMessage->amount;
    LOG("Checker %d was healed for %d by checker %d",
        GetID(), pMessage->amount, pMessage->healedBy);
    return JAM_OK;
}
```


Development Cycle

- Describe the protocol
- Generate serialization and dispatch
- Send messages
- Receive messages
- Configure routing info

Define services

```
void Matchmaker::Configure(JamServer *pServer)
{
    JamRouteConfig &routeConfig = pServer->GetRouteConfig();
    routeConfig.ConfigureInbound<MatchmakerProtocol>(
        this, Matchmaker::DispatchMessage);
    routeConfig.ConfigureOutbound<MatchmakerResponseProtocol>();
}
```

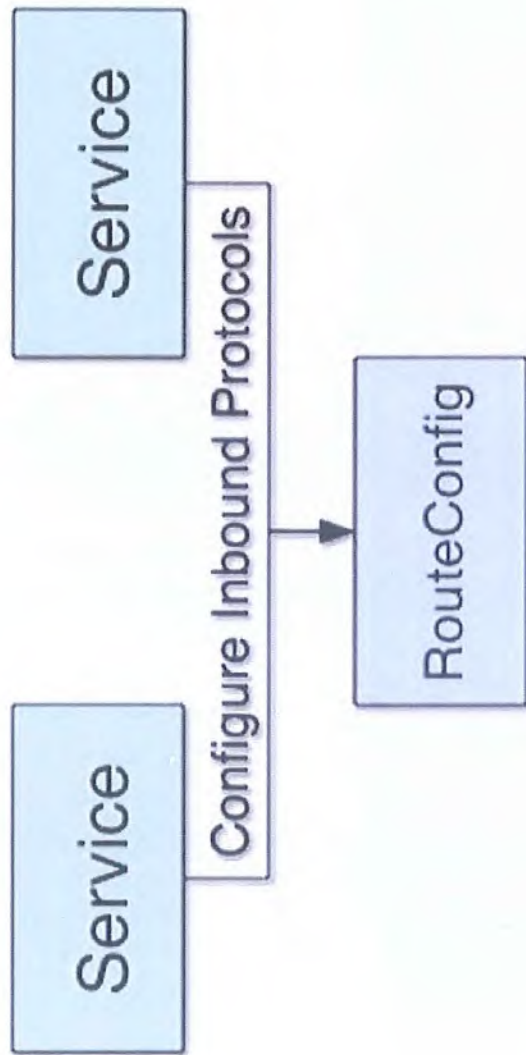
Configure protocols the Matchmaker service sends and receives

Define services

```
void Matchmaker::Configure(JamServer *pServer)
{
    JamRouteConfig &routeConfig = pServer->GetRouteConfig();
    routeConfig.ConfigureInbound<MatchmakerProtocol>(
        this, Matchmaker::DispatchMessage);
    routeConfig.ConfigureOutbound<MatchmakerResponseProtocol>();
}
```

Configure protocols the Matchmaker service sends and receives

RouteConfig maintains a protocol to handler mapping



Handlers have access to sender and other metadata about received messages

```
JAM_RESULT BoardServer::AddPlayerHandler(JamLink &Link,  
AddPlayer *msg)  
{  
    LOG("Adding player %s from server %s",  
        IDSTR(msg->playerID),  
        Link.Describe().c_str());  
    // Do stuff  
    return JAM_OK;  
}
```

Handlers have access to sender and other metadata about received messages

```
JAM_RESULT BoardServer::AddPlayerHandler(JamLink &Link,
                                           AddPlayer *msg)
{
    LOG("Adding player %s from server %s",
        IDSTR(msg->playerID),
        Link.Describe().c_str());
    // Do stuff
    return JAM_OK;
}
```


Coarse and fine-grained queuing and



Race Condition

Receiving via Message Queue

```
void Matchmaker::Configure()  
{  
    // Messages received at any time are placed into a queue  
    routeConfig.ConfigureInbound<MatchmakerProtocol>(  
        this, &m_messageQueue);  
}  
  
void Matchmaker::Idle()  
{  
    // Queue is processed in one thread at a known time  
    pServer->ProcessQueue(&m_messageQueue, this);  
}
```


Receiving via Message Queue

```
void Matchmaker::Configure()  
{  
    // Messages received at any time are placed into a queue  
    routeConfig.ConfigureInbound<MatchmakerProtocol>(  
        this, &m_messageQueue);  
}  
  
void Matchmaker::Idle()  
{  
    // Queue is processed in one thread at a known time  
    pServer->ProcessQueue(&m_messageQueue, this);  
}
```

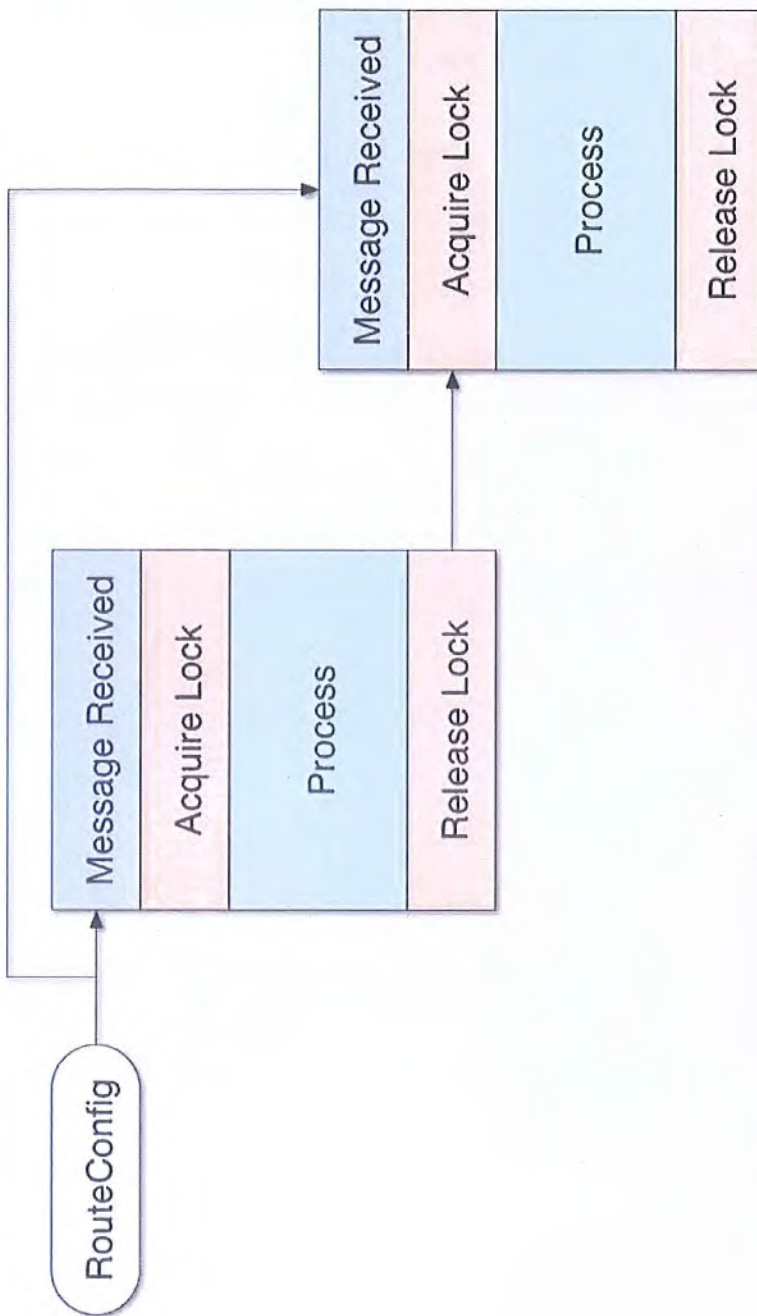

Receiving via Message Queue

```
void Matchmaker::Configure()  
{  
    // Messages received at any time are placed into a queue  
    routeConfig.ConfigureInbound<MatchmakerProtocol>(  
        this, &m_messageQueue);  
}  
  
void Matchmaker::Idle()  
{  
    // Queue is processed in one thread at a known time  
    pServer->ProcessQueue(&m_messageQueue, this);  
}
```

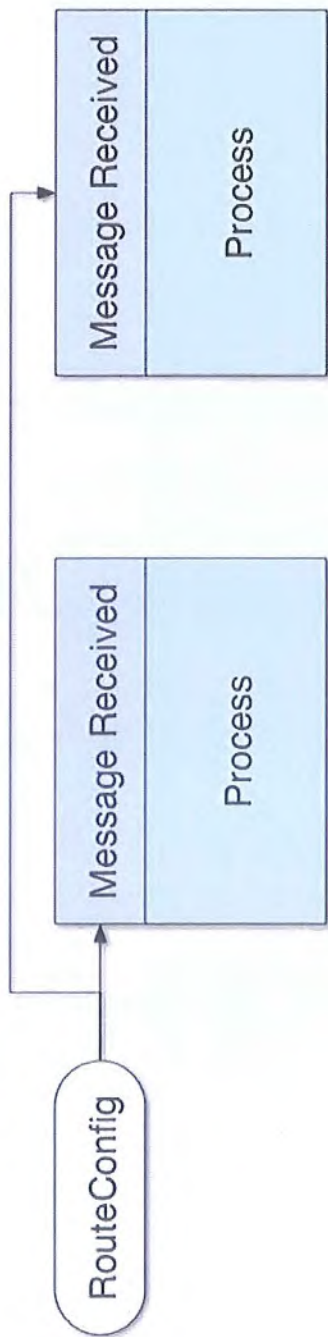
Receiving via Message Queue

```
void Matchmaker::Configure()  
{  
    // Messages received at any time are placed into a queue  
    routeConfig.ConfigureInbound<MatchmakerProtocol>(  
        this, &m_messageQueue);  
}  
  
void Matchmaker::Idle()  
{  
    // Queue is processed in one thread at a known time  
    pServer->ProcessQueue(&m_messageQueue, this);  
}
```

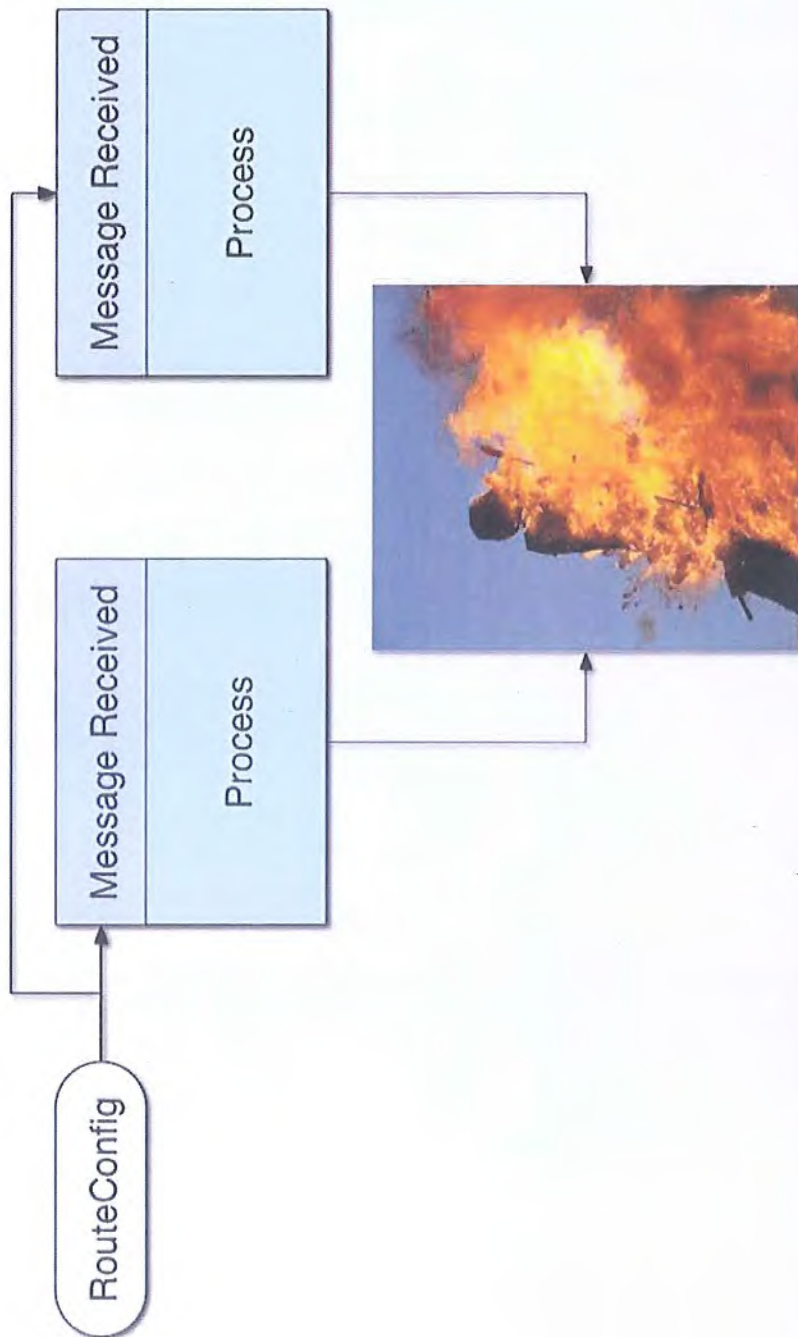

Global lock dispatching



Raw concurrent handlers



Raw concurrent handlers



Lock Policies

```
class MatchmakerLockPolicy
{
    Matchmaker *m_owner;
    void Lock(JamMessage *msg, JamMessageQueue **ppQueue)
    {
        // Adding a player requires a write lock
        if (msg->GetCode() == JAM_MSG_MatchmakerAddPlayer) {
            m_owner->AcquireWriteLock();
        } else {
            m_owner->AcquireReadLock();
        }
    }
    void Unlock(JamMessage *msg) { /* Same logic, release lock */ }
}
```



Lock Policies

```
class MatchmakerLockPolicy
{
    Matchmaker *m_owner;
    void Lock(JamMessage *msg, JamMessageQueue **ppQueue)
    {
        // Adding a player requires a write lock
        if (msg->GetCode() == JAM_MSG_MatchmakerAddPlayer) {
            m_owner->AcquireWriteLock();
        } else {
            m_owner->AcquireReadLock();
        }
    }
    void Unlock(JamMessage *msg) { /* Same logic, release lock */ }
}
```

Incoming messages are recounted

GOC¹³

WIZARD ENTERTAINMENT

AB-AB 002920

Incoming messages are recounted

- Message passed to handler is a recounted object

Incoming messages are recounted

- Message passed to handler is a recounted object
- Possible to retain a message pointer until later

Incoming messages are recounted

- Message passed to handler is a recounted object
- Possible to retain a message pointer until later
- Smart pointers are available

Incoming messages are recounted

- Message passed to handler is a recounted object
- Possible to retain a message pointer until later
- Smart pointers are available
- Messages contain no pointers to any other objects

Incoming messages are recounted

- Message passed to handler is a recounted object
- Possible to retain a message pointer until later
- Smart pointers are available
- Messages contain no pointers to any other objects
- No circular references are possible

CPU And Bandwidth Efficiency

GOC¹³

RIZARD
ENTERTAINMENT

AB-AB 002926

**JAM is either efficient or backwards
compatible**

2004 - Assumed binary compatibility

GDC
13

HIZARD
ENTERTAINMENT

Negotiation means dead-simple binary
serialization most of the time

GOC
13



In some cases, can just memcopy it onto
the wire

```
// This message could easily be memcopy'ed onto the wire
class CreateChecker : public JamMessage {
    /*** DATA START ****/
    u32 checkerType;
    u32 owner;
    /*** DATA STOP ****/
    // Code...
};
```

GOC¹³

BIZARD
ENTERTAINMENT

Generated code means easy
optimizations

GOC
13

WIZARD
ENTERTAINMENT

Generated code means easy optimizations

```
_encoder.Put("capturedCheckerID", capturedCheckerID);  
_encoder.Put("capturedBy", capturedBy);  
_encoder.Put("jumpType", jumpType);
```

Fallback to JSON Serialization



GDC
13



Fallback to JSON Serialization

- Switch to JSON serialization when binary CRC check fails

Fallback to JSON Serialization

- Switch to JSON serialization when binary CRC check fails
- Great for programmers

GOC
13

HIZARD
ENTERTAINMENT

Fallback to JSON Serialization

- Switch to JSON serialization when binary CRC check fails
- Great for programmers
- Way more expensive (CPU and Bandwidth)

Fallback to JSON Serialization

- Switch to JSON serialization when binary CRC check fails
- Great for programmers
- Way more expensive (CPU and Bandwidth)
- Never allowed on public facing protocols

Fallback to JSON Serialization

- Switch to JSON serialization when binary CRC check fails
- Great for programmers
- Way more expensive (CPU and Bandwidth)
- Never allowed on public facing protocols
- Even internally it's sometimes unreasonable

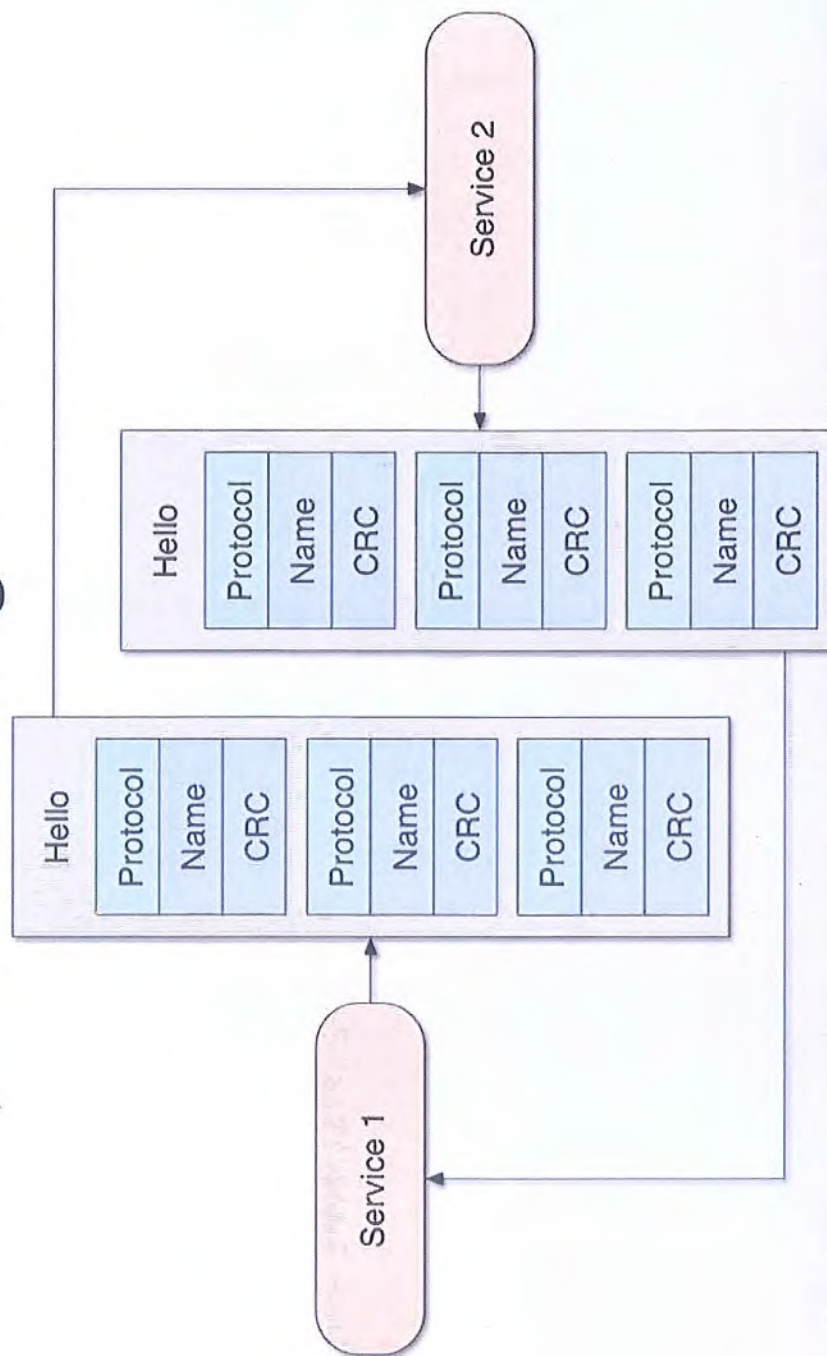
Fallback to JSON Serialization

- Switch to JSON serialization when binary CRC check fails
- Great for programmers
- Way more expensive (CPU and Bandwidth)
- Never allowed on public facing protocols
- Even internally it's sometimes unreasonable

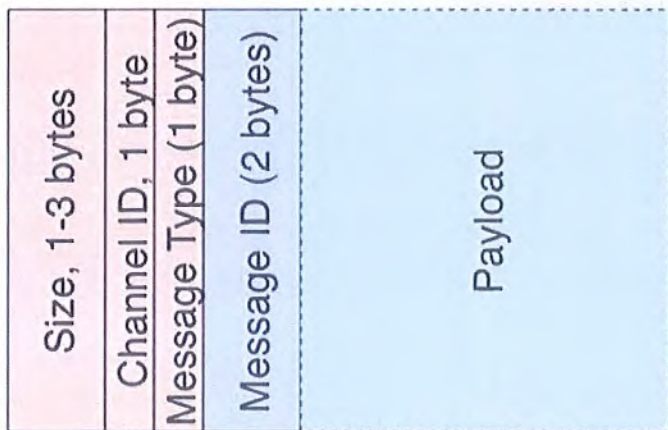
JSON

```
{
  "_msgID":10,
  "type":6,
  "error":0,
  "desc":{
    "m_id":"T2R00S40.00E14815726P10987H127.0.0.1:14001",
    "m_host":"127.0.0.1",
    "m_partitionID":0,
    "m_configID":0,
    "m_buildNum":0,
    "m_type":40,
    "m_subType":0
  }
}
```


Protocol Negotiation

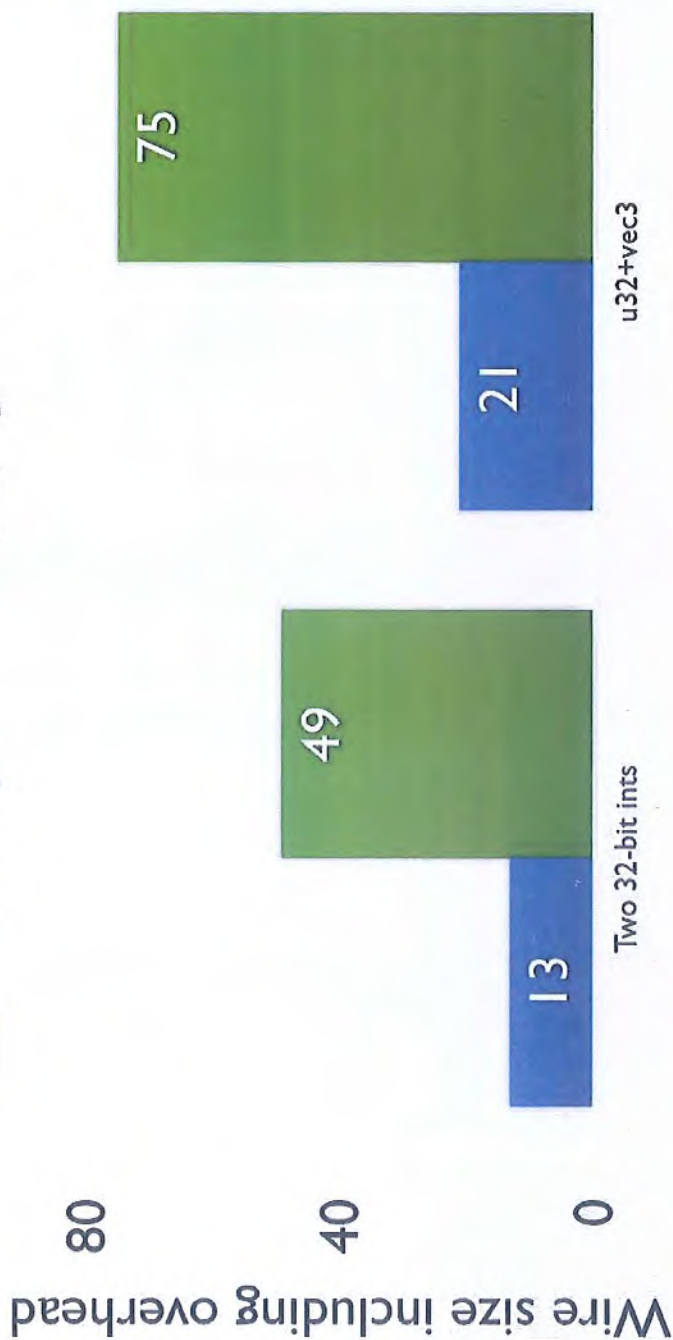


Message overhead



JSON vs. Binary performance

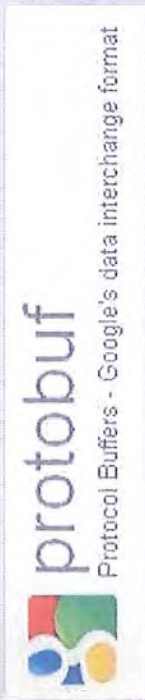
■ Binary ■ JSON



Still highly successful - some network tools run on old versions frequently

Builds	
Latest Build (8967)	
Blessed (8939)	
Latest Build (8967)	
8967	11/15 12:39 PM
8966	11/15 12:17 PM
8965	11/15 10:39 AM
8963	11/15 09:39 AM
8959	11/14 05:02 PM
8957	11/14 03:59 PM
8955	11/14 03:26 PM
8952	11/14 12:17 PM
8950	11/14 11:19 AM
8949	11/14 10:24 AM
8947	11/14 05:34 AM
8944	11/13 05:03 PM
8943	11/13 03:33 PM
8942	11/13 03:10 PM

Google's Protocol Buffers and



GOC
13

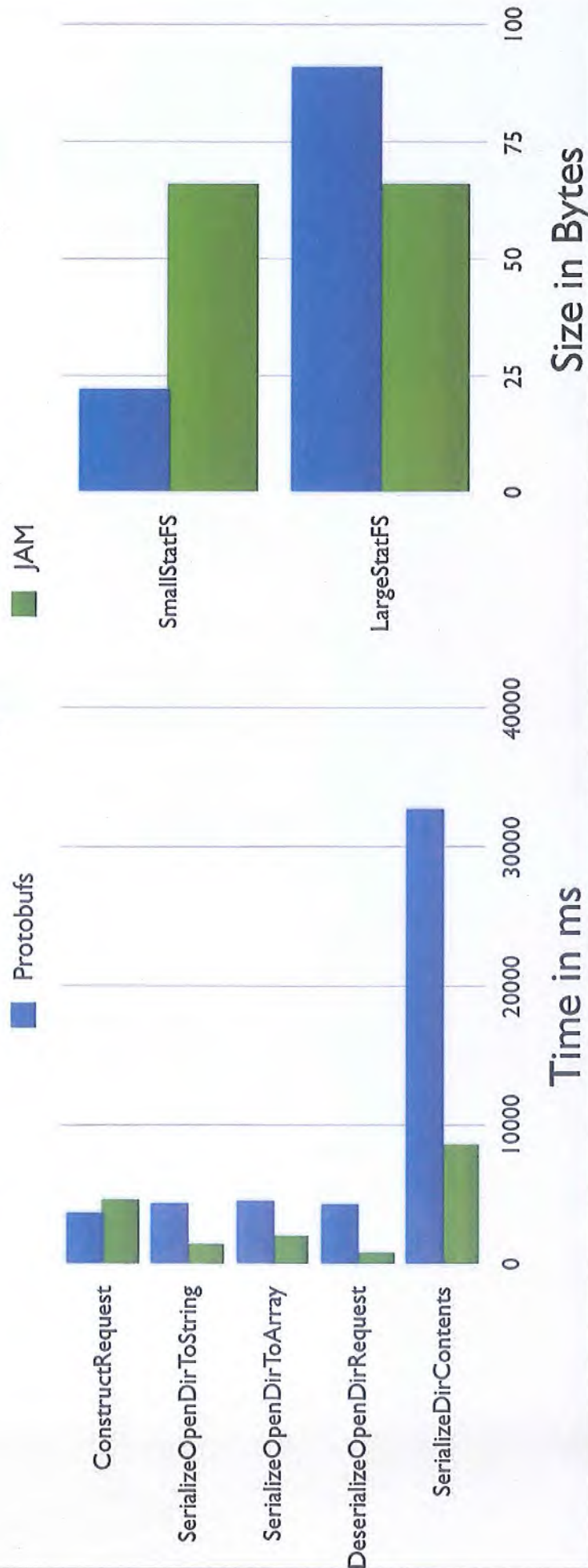


For both speed AND inter-version
compatibility, there are better choices

GOC¹³



protobuufs sometimes wins on bandwidth, but JAM is faster



**Writing our own gives us ultimate
control over everything**

GDC
13

WIZARD
ENTERTAINMENT

**Writing our own gives us ultimate
control over everything**

- Automated serialization

GOC
13

BUZZARD
ENTERTAINMENT

Writing our own gives us ultimate control over everything

- Automated serialization
- Easy yet flexible message dispatching

Writing our own gives us ultimate control over everything

- Automated serialization
- Easy yet flexible message dispatching
- High performance

Writing our own gives us ultimate control over everything

- Automated serialization
- Easy yet flexible message dispatching
- High performance
- Inter-version compatibility

GDC
13

RUZAN
ENTERTAINMENT

Writing our own gives us ultimate control over everything

- Automated serialization
- Easy yet flexible message dispatching
- High performance
- Inter-version compatibility
- Less tedium = more awesome

Thanks! Questions?

Joe Rumsey

jrumsey@blizzard.com

Twitter: @joerumz

GOC¹³

BLIZZARD
ENTERTAINMENT

Thanks! Questions?

Joe Rumsey
jrumsey@blizzard.com
Twitter: @joerumz

Disclaimer: Blizzard is not really making World of Checkers



EXHIBIT 15

**THIS EXHIBIT HAS BEEN
REDACTED IN ITS ENTIRETY**