US 20030023774A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2003/0023774 A1**

Gladstone et al. (43) **Pub. Date: Jan. 30, 2003**

(54) **STATEFUL REFERENCE MONITOR**

(76) Inventors: **Philip J. S. Gladstone**, Framingham, MA (US); **Jeffrey A. Kraemer**, Wellesley, MA (US)

Correspondence Address:
WOLF GREENFIELD & SACKS, PC
FEDERAL RESERVE PLAZA
600 ATLANTIC AVENUE
BOSTON, MA 02210-2211 (US)

(21) Appl. No.: **10/071,328**

(22) Filed: **Feb. 8, 2002**

Related U.S. Application Data

(60) Provisional application No. 60/298,590, filed on Jun. 14, 2001.

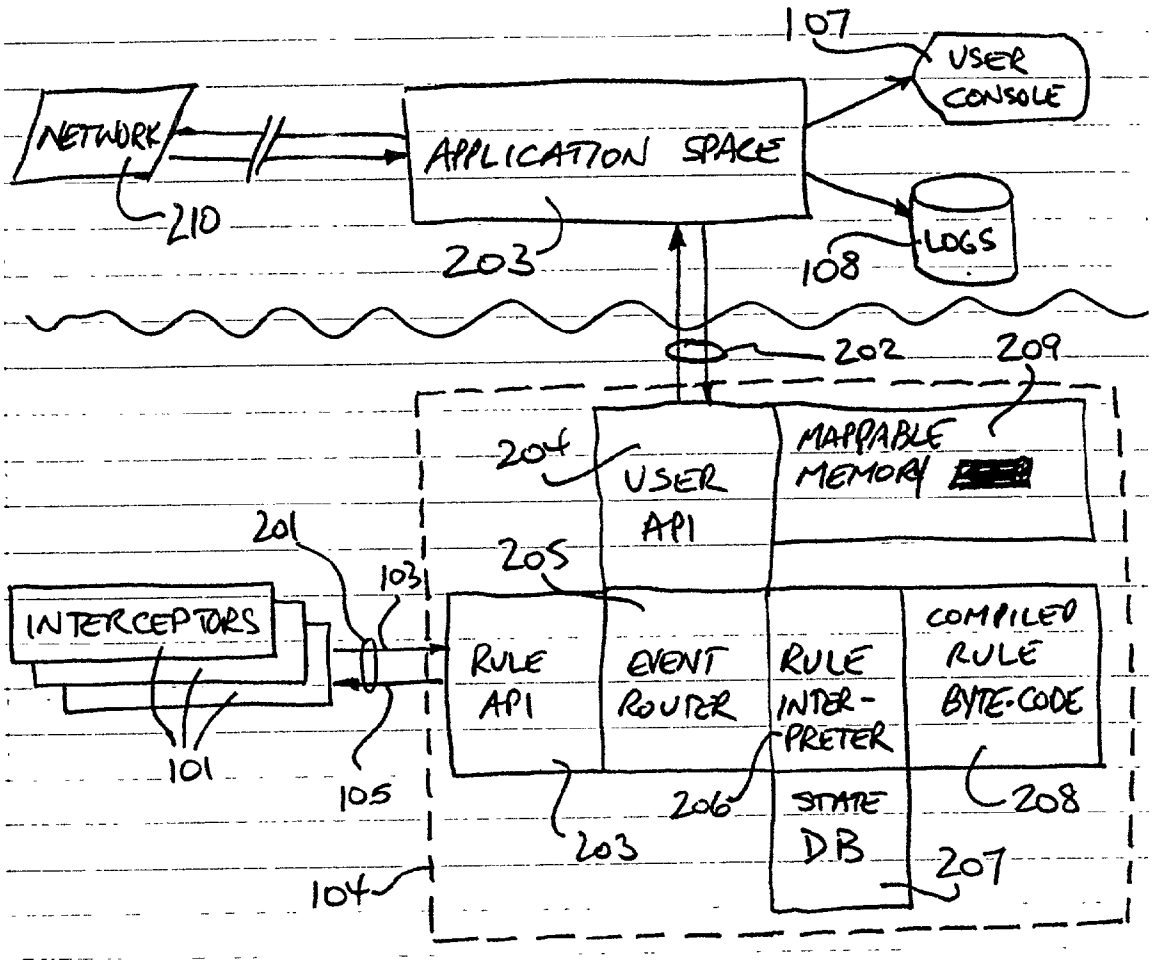
Publication Classification

(51) **Int. Cl.**[7] **G06F 9/00**

(52) **U.S. Cl.** **709/328**

(57) **ABSTRACT**

A Stateful Reference Monitor can be loaded into an existing commercial operating system, and then can regulate access to many different types of resources. The reference monitor maintains an updateable storage area whose contents can be used to affect access decisions, and access decisions can be based on arbitrary properties of the request.



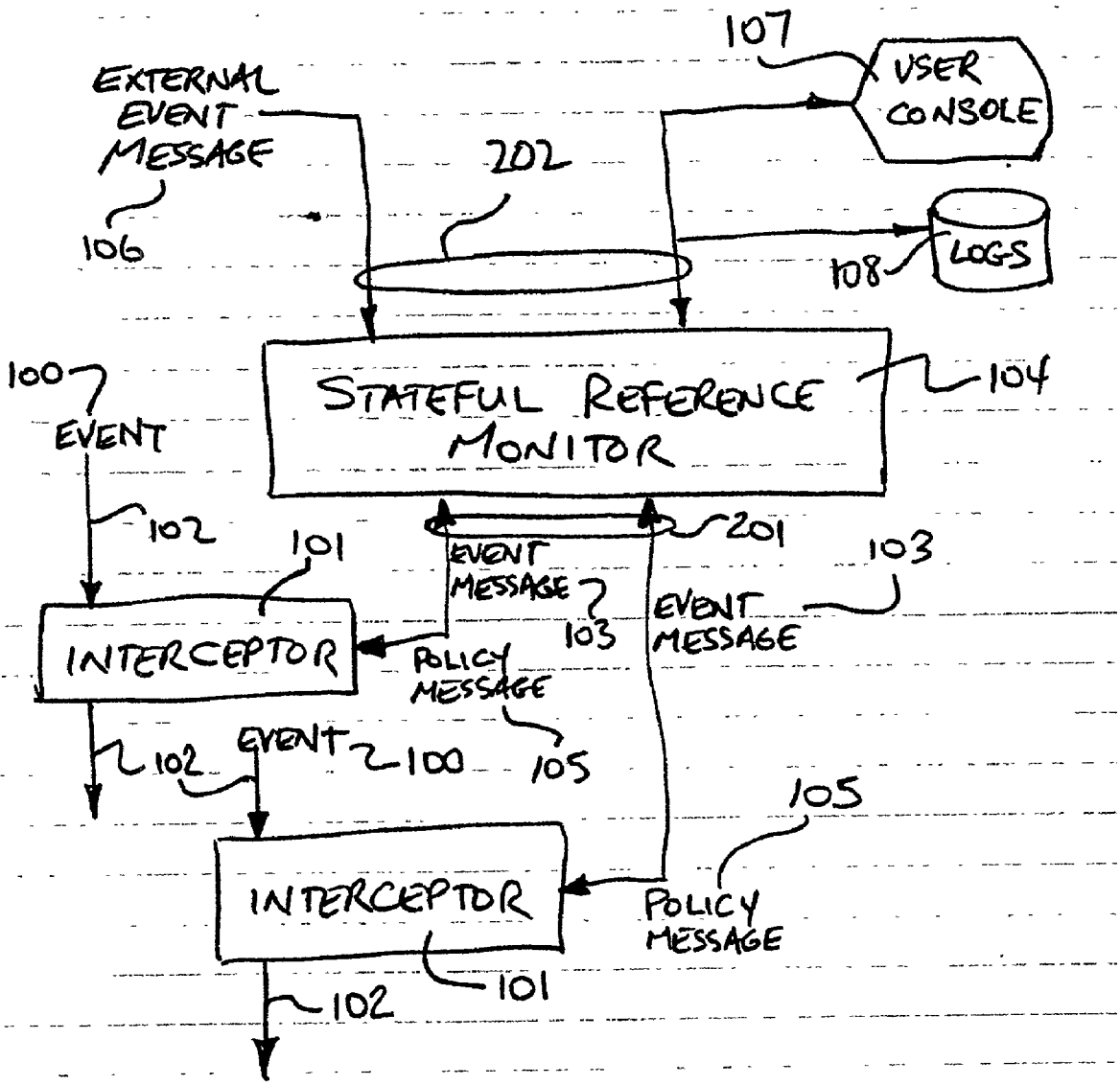


FIG. 1

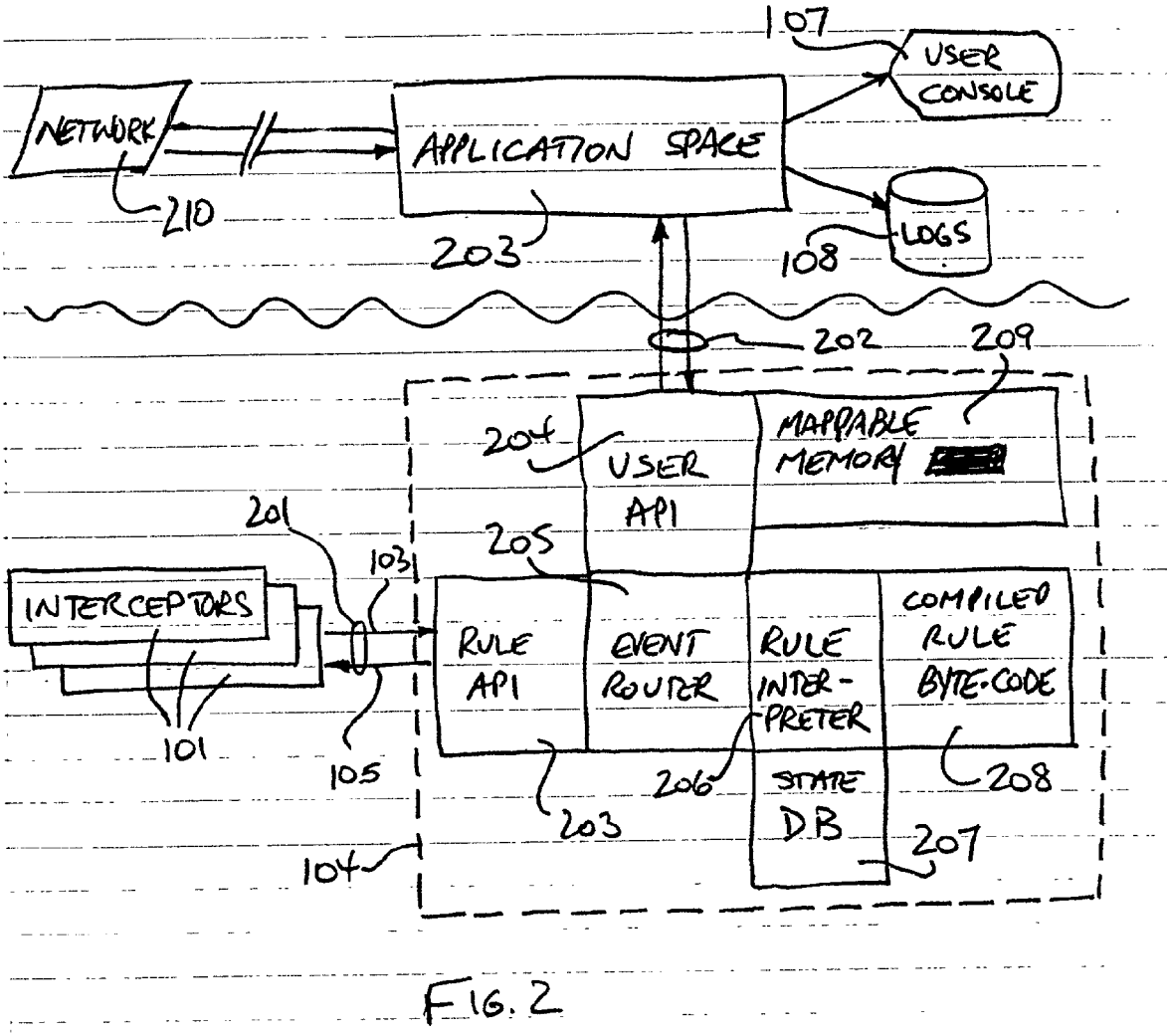


FIG. 2

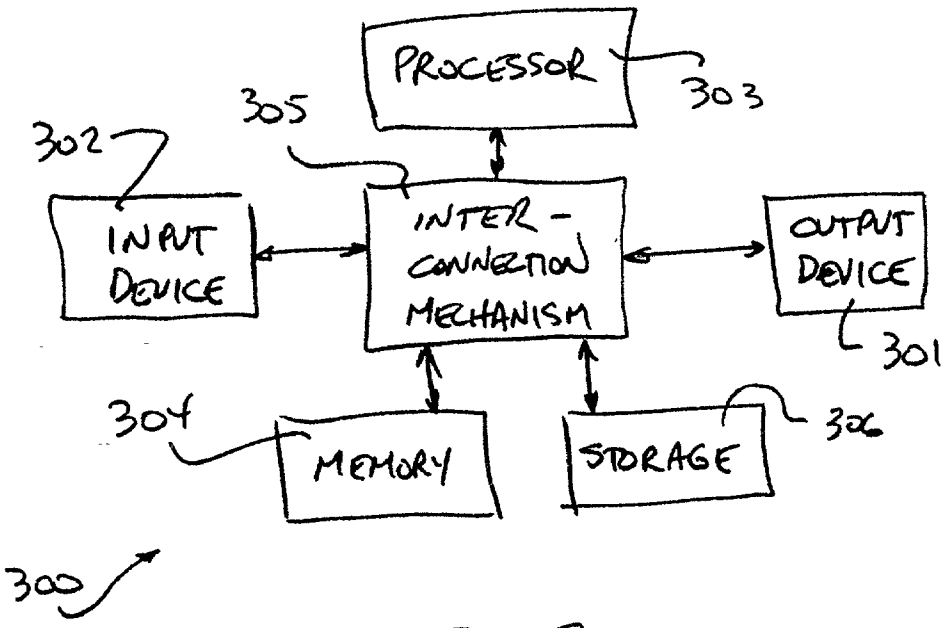


FIG. 3

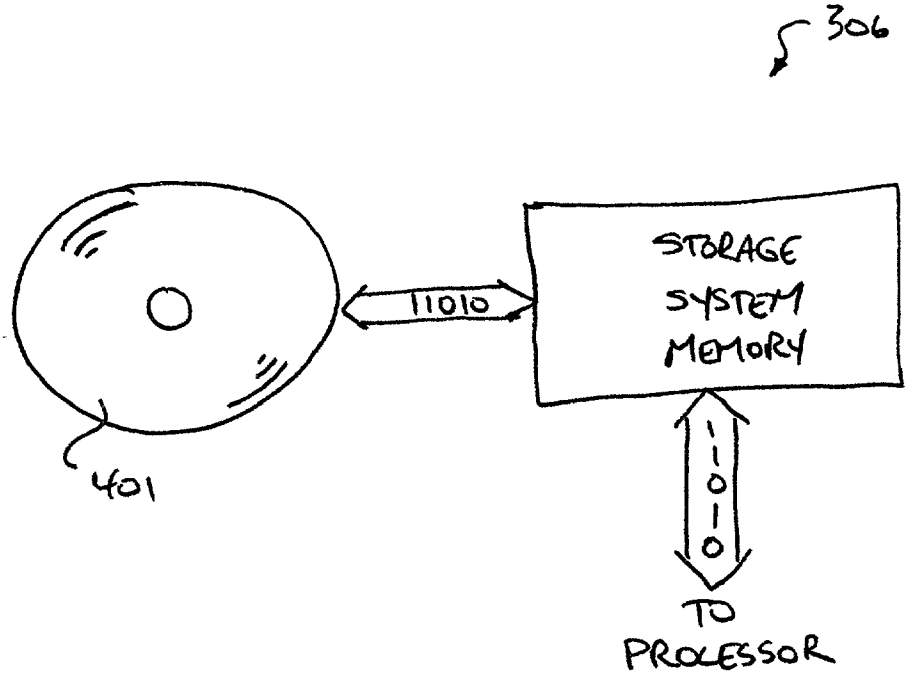


FIG. 4

STATEFUL REFERENCE MONITOR

CROSS-REFERENCE TO RELATED APPLICATION

[0001] The present application claims domestic priority under 35 U.S.C. §119(e) to copending U.S. provisional patent application serial No. 60/298,590 filed Jun. 14, 2001.

BACKGROUND OF THE INVENTION

[0002] The present invention related generally to software that controls an operating policy of a computer system. For example, access to various system resources, such as files and network connections may be so controlled.

[0003] Modern computer systems are controlled by two levels of software: an operating system and application software. The operating system maintains a separation in the memory of the computer between the operating system, said to execute in operating system space, and the applications, said to execute in applications space.

[0004] Applications are the software that provides solutions to business problem, such as processing e-mail messages. Applications perform their work by communicating requests for access to resources such as network connections or files stored on a storage medium to the operating system, which then fills these requests.

[0005] It may be desired to control any one or more of a wide variety of operating policies. One common scenario is to control access to various system resources, as mentioned above, for purposes of securing a system against deliberate and malicious attack or for purposes of ensuring correct and non-interfering operation of various processes. For purposes of illustration, access control for security purposes is discussed. However, the skilled artist will understand that the discussion has wider implications, as explicitly pointed out and as implied in the following discussion.

[0006] Referring to the example of security systems, access is conventionally controlled by one or more real-time processes, while policy is independently established with the assistance of one or more non-real-time processes. In the context of this description, a real-time process is one whose action is sufficiently immediate as to imperceptibly affect the speed at which transactions with which the real-time process is connected are processed. A non-real-time process is one that processes transactions at a speed substantially slower than the instantaneous rate at which transactions naturally occur.

[0007] Real-time access control processes and data structures include, but are not limited to reference monitors, access control lists, permissions flags, access tokens and process ID checking.

[0008] A reference monitor is a component of a computer system that determines whether an access by one component, for example a user process, of another component, for example, a file is permitted.

[0009] As used hereinafter, dynamic state is a collection of information, that is collected in real-time, indicative of a condition of a machine or process as a result of a particular sequence of events leading to the condition. A stateless system or component is one, which does not collect such data.

[0010] Conventional reference monitors, herein referred to as stateless reference monitors, are found in the kernels of various Operating Systems, including, for example, MicroSoft® Windows™ 2000 OR UNIX. They are used to determine whether a particular access to a file or other resource is permitted.

[0011] Conventional operating systems contain embedded stateless reference monitors to control access to resources. User processes are started and identified to users on the basis of the user supplying certain identity tokens. In most cases the access decision is made based on the identity of the user whose local program or process makes the request and one or more static permissions flags or an access control list associated with the resource. For examples, see Unix or Windows 2000. The contents of static permissions flags and access control lists do not include information representing the current state of the system, but rather include information that produces identical results regardless of the state of the system.

[0012] Most conventional reference monitors deals with a single resource type (such as files or network connections). Some, such as eTrust Access Control v.5.1 from Computer Associates, protect multiple resource types.

[0013] Some operating systems give finer control by associating individual permissions with each user, and then checking those permissions against the static access control list of the resource. This is an improvement, but typically there are only a limited number of permission flags. Security-Enhanced Linux is an example of such an operating system.

[0014] There are operating systems that are even finer grained, and allow individual users to offer a set of tokens, and if any match those found in the access control list, then access is granted.

[0015] There are operating environments that can include the origin of the requesting program in their access control decision. For example, see Dan Wallach and Edward Felton, "Understanding Java Stack Inspection", IEEE Proceedings of Security & Privacy, May 1998.

[0016] Non-real-time processes are conventionally employed to collect date and analyze past events in order to establish or modify effective policies, for example security policies. Typical, conventional non-real-time processes include intrusion detection systems, for example.

[0017] One type of intrusion detection system is an autonomous agent that polls, monitors and/or periodically mines log files for data indicative of an intrusion. A drawback of such non-real-time systems is that intrusions are only detected "after the fact." The intruder leaves an audit trail of actions recorded in log files and elsewhere for which the only reasonable explanation is an intrusion. However, by the time such a non-real-time intrusion detection system identifies an intrusion, the intruder is long gone and damage done.

[0018] For examples, see Peter G. Neumann and Phillip A. Porras, "Experience with EMERALD to Date", 1st USENIX Workshop on Intrusion Detection and Network Monitoring, April 1999; Eugene Spafford et al. "Intrusion detection using autonomous agents" Computer Networks 34 (2000); and Steven R. Snapp et al., "DIDS (Distributed Intrusion Detec-

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.