

1. Introduction

Telesma Sentinel (TS) provides distributed monitoring of multiple systems for error or warning conditions, producing alerts and notifying users of the need to take action. It is a tool that facilitates system and database administration by automating the tedious tasks of periodically checking operating systems, application processes, filesystems, logs, databases, and network elements to make sure that they are all working properly and have not encountered or are about to encounter a problem situation such as lack of disk space.

2. Problem Statement

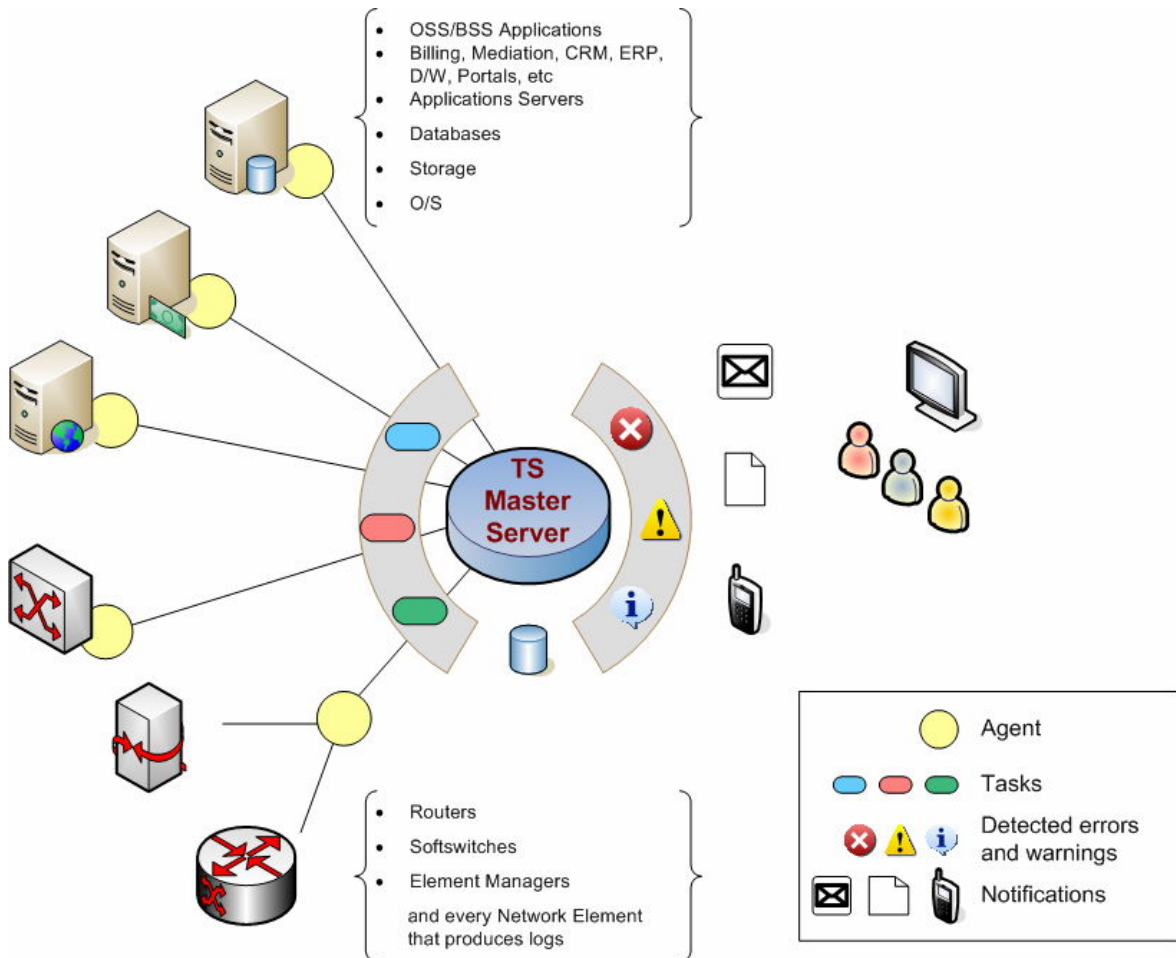
In a typical IT infrastructure, one or more personnel of various roles need to periodically monitor the infrastructure for potential problems in order to prevent failures or minimize the impact of such failures in case they occur, by detecting them early.

An IT organization typically has multiple areas of responsibility that are handled by different people with roles such as: Database Administrators, System Administrators, Application Administrators and Operators. Each of these people is called to monitor an area of responsibility. A database administrator makes sure that the database has the disk space that it needs, that it is running properly, and that it is tuned up. A system administrator makes sure that the system is up, that the filesystems are not about to fill up, and that certain core processes are running. An application administrator makes sure that the application processes are running, that the application is making progress processing data, and that there are no errors reported in the application's logs.

3. Solution

Telesma Sentinel provides a distributed, centrally-configurable flexible system that IT administrators can use to setup automated monitoring tasks. Each task is scheduled to monitor one aspect of one target host. It runs at a configured schedule and reports any conditions that need attention (alerts). The **TS** collects all alerts and sends notifications via email or other means, so that the appropriate person can take action.

4. Architecture



Telesma Sentinel Architecture

Telesma Sentinel has a distributed architecture that consists of:

- an agent installed on each system that should be monitored
- a **TS** master server that communicate with all agents and orchestrates the execution of tasks and the dispatching of notifications
- a centralized configuration that is used to define the monitoring tasks, their schedules, and the notifications
- a Web-based GUI that lets users administer the **TS** configuration and view reports

5. Features

- **Distributed Monitoring:** **TS** monitors multiple systems. A lightweight system monitoring agent is installed in each monitored system. A central system monitor server contacts the agents, sends monitoring tasks to them, collects the results and sends the notifications.
- **Central Configuration:** All tasks of a multi-system **TS** installation are stored and configured in a central location. A user can test a monitoring task from this central location, without needing to log on to each system separately to set it up.
- **Multiple Users:** **TS** has its own user management for controlling access. It lets users configure their own monitoring tasks and notifications independently of each other, so that each user is responsible and accountable for their own aspects of monitoring.
- **Web GUI:** **TS** is configured via a web application. The user interface is accessed through any standard web browser. The GUI makes it easy to configure and schedule a monitoring task and test it with the click of a button.
- **Reusable Tasks:** **TS** comes with a library of general purpose monitoring task types. These tasks encapsulate Telesma's expertise so that they can be used in a variety of situations and systems. Telesma will work with a customer in providing an initial configuration of tasks that suits the customer's environment. Telesma may also create additional custom tasks to take care of any special monitoring that the customer needs.
- **Extensibility:** **TS** uses a pluggable framework for defining task types. Task types can be written in Java and added to **TS** without changing any existing code. Telesma can work with the customer in incorporating any existing monitoring applications that the customer may have. For example, tasks may be written in any language as UNIX programs that output alerts to standard output or standard error and can be called by **TS** via a Java bridge task.

6. Benefits

- **Holistic View:** **TS** provides a holistic view of a business's information processes. It monitors multiple aspects in multiple systems with a final focus on the business mission.
- **Continuous Monitoring:** Once **TS** is set-up, it provides continuous unattended monitoring. It drastically reduces the tedious repetitive part of monitoring and allows

the administrators to focus on the more creative work of attending to the business needs of their organization instead of baby-sitting systems.

- **Easy Configuration:** If a monitoring procedure becomes out-of-date and produces incorrect notifications then it loses its value. **TS** is designed with the assumption that the need of the business constantly changes. It therefore allows the users to easily modify the monitoring configuration, by adding agents, adjusting monitoring tasks, and changing schedules.
- **Improved Quality of Service:** The bottom line of **TS** is that its use improves the quality of an organization's IT operations by reducing failures and addressing problems proactively.

7. Monitoring Tasks

This section briefly describes a few representative monitoring tasks that are addressed by **TS**. This list is given to illustrate the capabilities of the system and the needs that it addresses.

Certain processes are running and vice-versa some others are not running

For example, Oracles' core processes should be alive, but the db's recovery process should be not active

The system has enough disk space to operate

Verify that a list of filesystems is available, and that each filesystem has a minimum amount of free space, as determined by a threshold either as a percentage of space used, or an absolute free size in Kb

Log files should be monitored for errors

Verify that log files don't have errors, or that they have expected maintenance events. Specify log entries of interest using regular expressions. The log monitoring task works incrementally so it shows only the log entries since the last time it was run

The database is in good health

Verify that a list of tablespaces in a database have enough free space, that indexes are analyzed, and that a database is online

The application process data

Verify that data files are added or removed in an input data directory or verify that a database table has new rows since the last time it was checked

Business Rules are followed

For example, in the uFM System all the customers in the History Table should have "Terminated" status