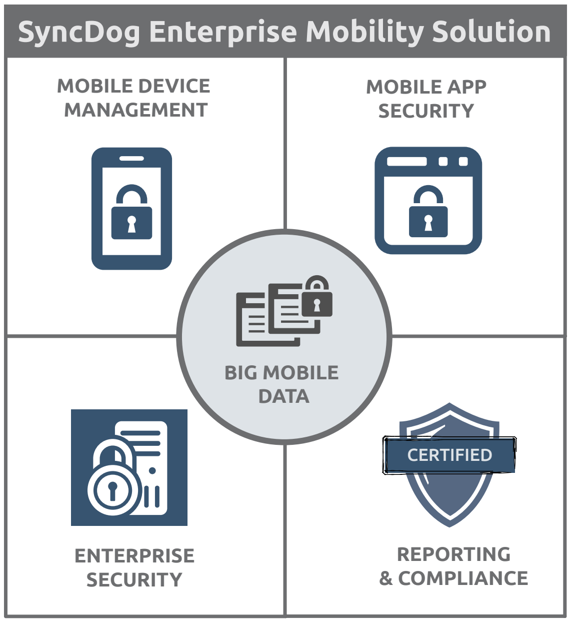
Sentinel Secure Server Structure: 4 Implementation Scenarios

Sentinel Secure is a component of SyncDog's Enterprise Mobility Solution Suite that helps large enterprises with mobile workforces manage mobile device provisioning, security and compliance. The main function of Sentinel Secure is to provide mobile application security through the use of secure containers. Sentinel Secure allows your organization to control and manage how your people exchange relevant, time-critical information across tablets and wireless handheld devices. This allows your IT team to establish and enforce secure enterprise-wide policies to protect everyday collaboration and communication. Sentinel Secure leverages AES 256-bit encryption to keep business data and apps in containers that are completely under the control of the organization's IT, without impacting the personal side of the device.



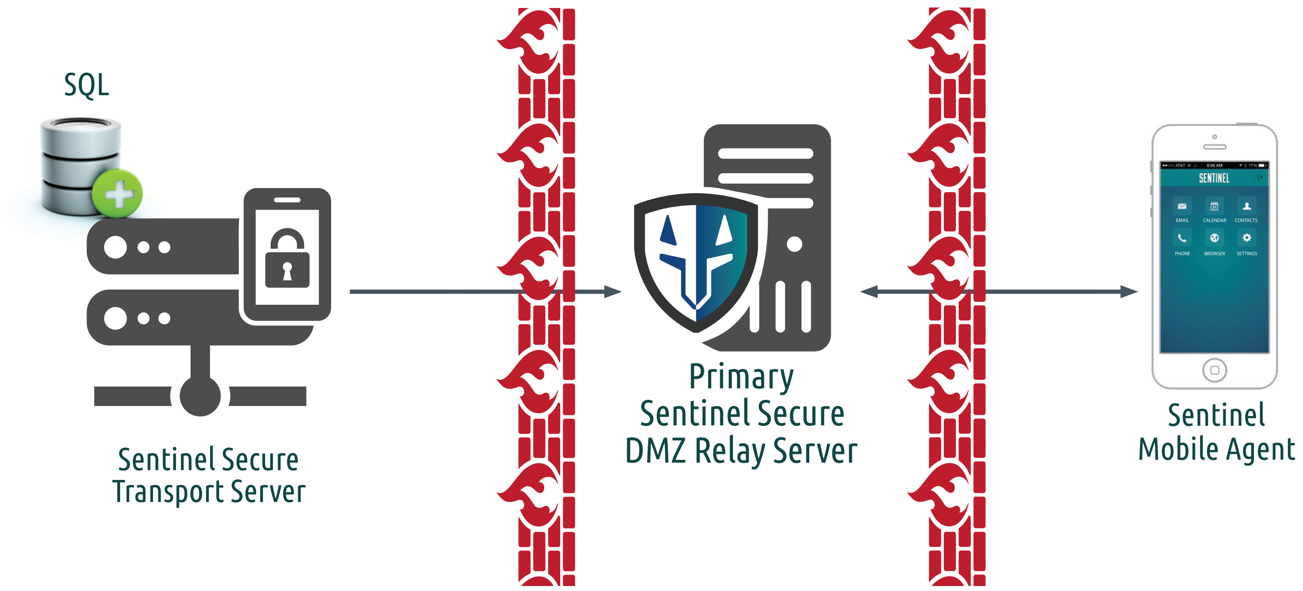
There are two different versions of Sentinel Secure's containerized solution:

- Client/server version administered by a systems admin
- Client/server version with and SDK/API

This document provides four different implementation scenarios for Sentinel Secure.

The following is the most basic implementation of Sentinel Secure possible:

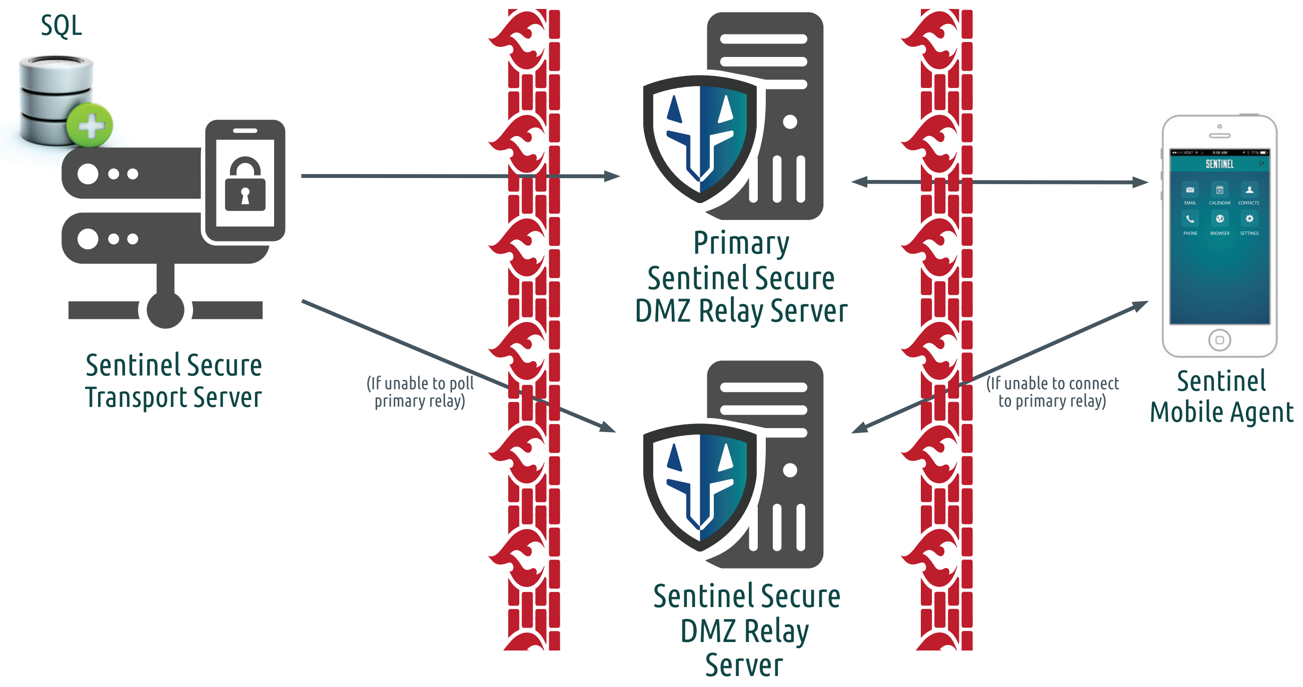
This model contains a single relay server, and a single transport server. The SQL Server database exists on the transport server.


SQL
Sentinel Secure Transport Server
Primary Sentinel Secure DMZ Relay Server
Sentinel Mobile Agent

Scenario 1: Sentinel Secure Basic Implementation

Level 1 - Relay Scalability and Failover

Because of the highly dynamic state of the relay messages and the speed at which they are processed, relay message queues are not persisted across multiple servers. Relay scalability is achieved primarily by device-side policy. When a user provisions, part of the user policy and configuration package that is sent to the device will include a list of relays along with a connection priority value. When a device wants to initialize a connection to the server, it will attempt to connect to the highest priority relay first, and if it gets a valid response from the server allowing to connect and initialize, it will continue to use that relay for subsequent requests. If it does not get a valid response from the relay, or if the attempted connection was disallowed because the relay responds and says it is too busy to accept new connections, the device will try to connect to the next highest priority relay, and so on. Sentinel Secure also provides policy rules in the user setup process that allows people on the west coast to have their own set of relays while east coast users can have their own set of relays.


SQL
Sentinel Secure Transport Server
(If unable to poll primary relay)
Primary Sentinel Secure DMZ Relay Server
Sentinel Secure DMZ Relay Server
(If unable to connect to primary relay)
Sentinel Mobile Agent

Scenario 2: Sentinel Secure Dual Relay Implementation

This scenario requires relays to be performance aware, and to know when they are too busy to accept new connections. SyncDog's position is to have a three part approach to this issue:

1. As part of the relay configuration, a system administrator could be able to set a max number of concurrent connections, after which the relay would refuse connections until the active connection count drops back below this threshold.
2. The relay should monitor its own performance internally and reject new connections at a certain threshold of RAM or constant CPU usage.
3. The relay essentially has one job: manage the inbound and outbound queue of messages.

Error conditions such as complete hardware failures or unresponsive relay servers will cause both the device attempting to connect and the backend transport servers that are polling the relay to roll over to their next highest priority relay server.

Unless there is a physical problem with the server or an inordinate number of transport servers all pulling from the same relay, performance bottlenecks will likely not be in the relay being able to queue the messages, but rather in the transport servers' ability to process the messages. Therefore, it is necessary for the transport server to also be able to tell the relay servers that they are unavailable to process any more messages. This will allow the relay to either queue the request for another transport server, or reject any more incoming connections and allow the device to connect to the next relay in their policy.

Level 2 – Transport Scalability and Failover

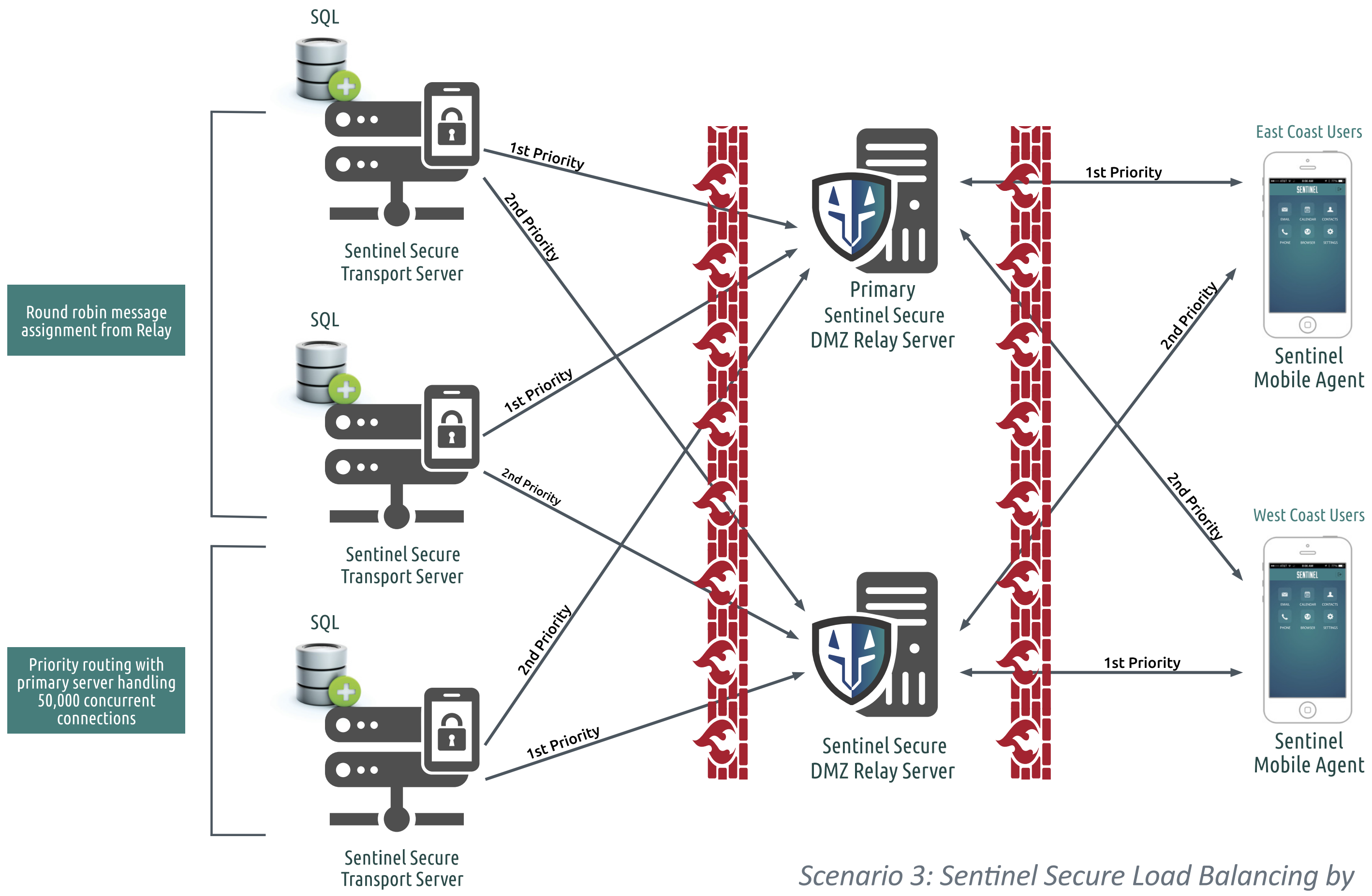
As previously stated, the transport servers do the bulk of message processing; so these machines are at the greatest risk of causing performance bottlenecks. Transport servers can be configured for load balancing and failover in two ways:

1. As with the relay, a system administrator should be able to configure a static number of maximum connections that are allowed to be concurrently running on the transport servers.
2. The servers should be self-monitoring to refuse connections based on RAM usage and constant CPU performance.

Once the failover thresholds have been crossed, the transport server would send a message to the relay telling the relay to pause queuing connections for it until it gets another message that allows it to respond to messages again.

TRANSPORT LOAD BALANCING

Since a transport server is self-monitoring and informs the relay when it is no longer able to handle new connections, it is possible to configure a transport server "pool." For example, three transport servers could all be pulling from the ▷

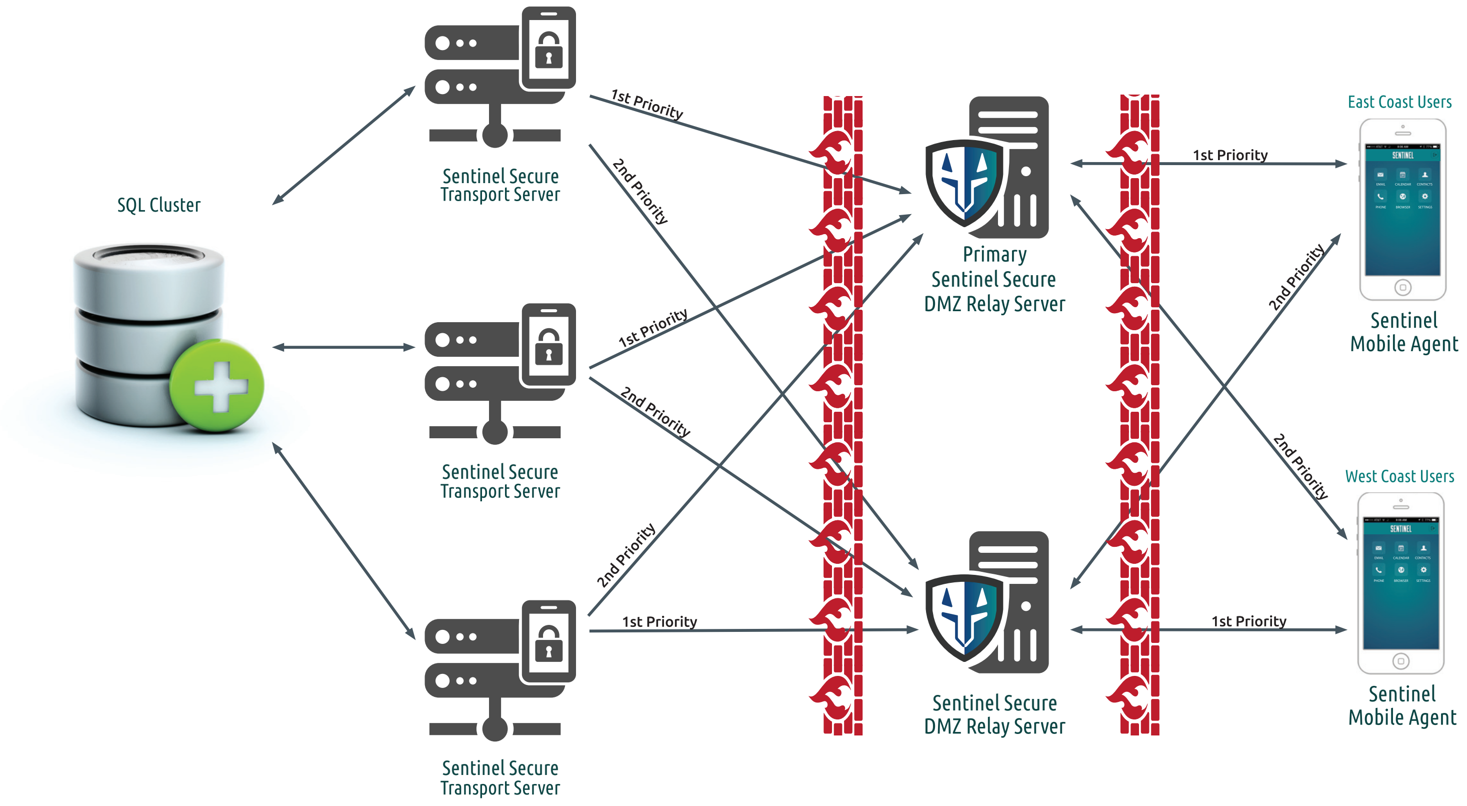

SQL
Sentinel Secure Transport Server
SQL
Sentinel Secure Transport Server
SQL
Sentinel Secure Transport Server
Round robin message assignment from Relay
Priority routing with primary server handling 50,000 concurrent connections
1st Priority
2nd Priority
1st Priority
2nd Priority
2nd Priority
1st Priority
Primary Sentinel Secure DMZ Relay Server
Sentinel Secure DMZ Relay Server
1st Priority
2nd Priority
2nd Priority
1st Priority
East Coast Users
Sentinel Mobile Agent
West Coast Users
Sentinel Mobile Agent

Scenario 3: Sentinel Secure Load Balancing by Priority Implementation same relay. A system administrator should be able to configure the routing behavior of the pool in the transport server configuration. If the pool is configured for round-robin routing, the transport servers would all register with the relay on startup letting it know they are available to receive connections and process messages. The relay would then queue incoming connections addressed to each server in rotation. The transport relay queue polling process would then pick up only messages are addressed to it from the relay.

A second available configuration option for transport load balancing is by priority routing. In this scenario, an administrator could specify for one transport server to handle all connections until it reaches its maximum connection or performance threshold, at which time another transport server would pick up processing connections.

Level 3 – Data Scalability and Failover

Whatever transport configuration is chosen by the system administrator, a current user and policy database must be shared between all the transport servers. The following diagram is a common scenario for achieving this goal. Administrators would install a SQL Server cluster to which all the transport servers connect to interact with data. Data access and redundancy would be handled internally in SQL Server and would be abstracted away from Sentinel Secure setup. All transport servers would simply connect to the cluster and retrieve their data.



Scenario 4: Sentinel Secure Load Balancing by Priority Implementation with SQL Synchronization

Sentinel Secure Features

- Device containerization with encrypted corporate workspace with IT-managed access controls, usage policies and remote commands
  - » E-mail with S/MIME and CAC
  - » Calendar and Contacts
  - » Office and PDF Document Suite with view, create, edit, annotate
  - » File Manager
  - » Secure Browser and Camera
  - » Application wrapper SDK
- Supports Android and iOS
- Defense-Grade Secure Workspace for Corporate Data Security & Compliance
  - » FIPS 140-2 AES 256-bit data encryption
  - » Hardware-separated Multi-factor Authentication (MFA)
  - » Resilient no-NOC architecture
  - » Support for S/MIME
  - » Proxy server behind-the-firewall
  - » Secure connector for ActiveSync
  - » Secure connector for browser and apps
- Ensures organizations can prove compliance in an auditable fashion.
- Full-featured clients for Microsoft Exchange with support for HTML e-mail, attachments and OTA calendaring

SyncDog, Inc.
1818 Library Street, Suite 500
Reston, VA USA 20190
Call: (703) 430-6040 | Fax: (703) 997-8667
sales@syncdog.com • www.syncdog.com