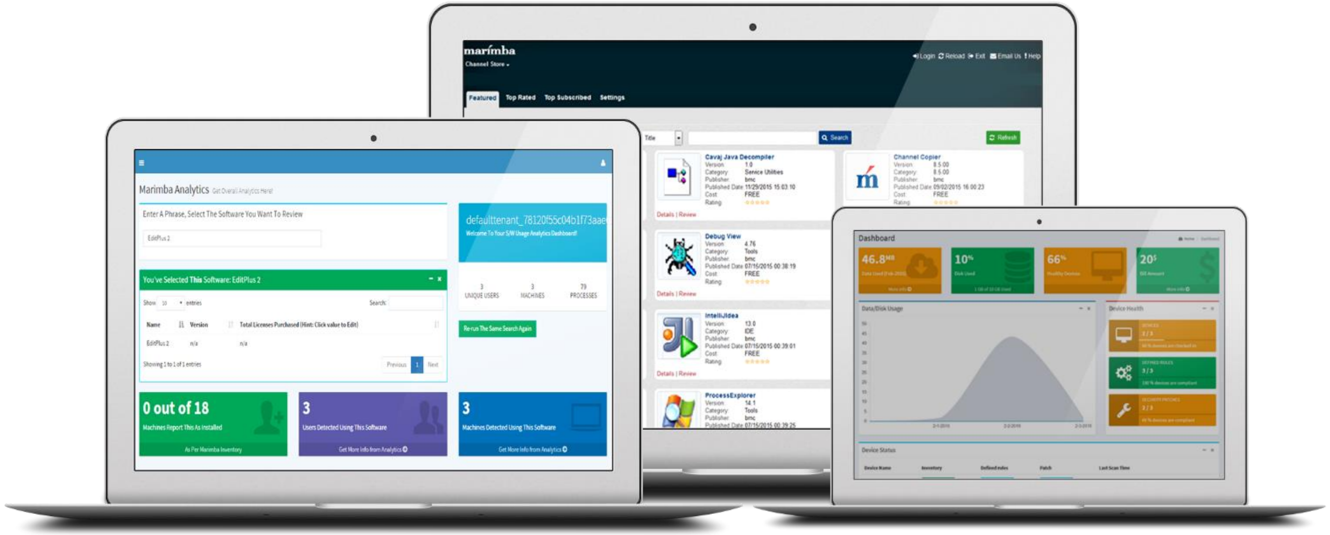


Marimba User Group

November 2017

Nitish Shrivastava
Product Manager & Chief Architect, Marimba

Castanet Logsight
- a quick recap

Common problem…

- Serious dependency on Inventory
  - Patch rollout
  - Policy updates
  - Machines activity

- Inventory scan-transport-insert delays reporting

- Aggressive Inventory would eat resources, choke plugin queue and impact DB performance

- There is a need of a parallel reporting framework that bridges this gap and offers close to real-time monitoring ability of critical deployments, patching and health of agents

How does the feature work?

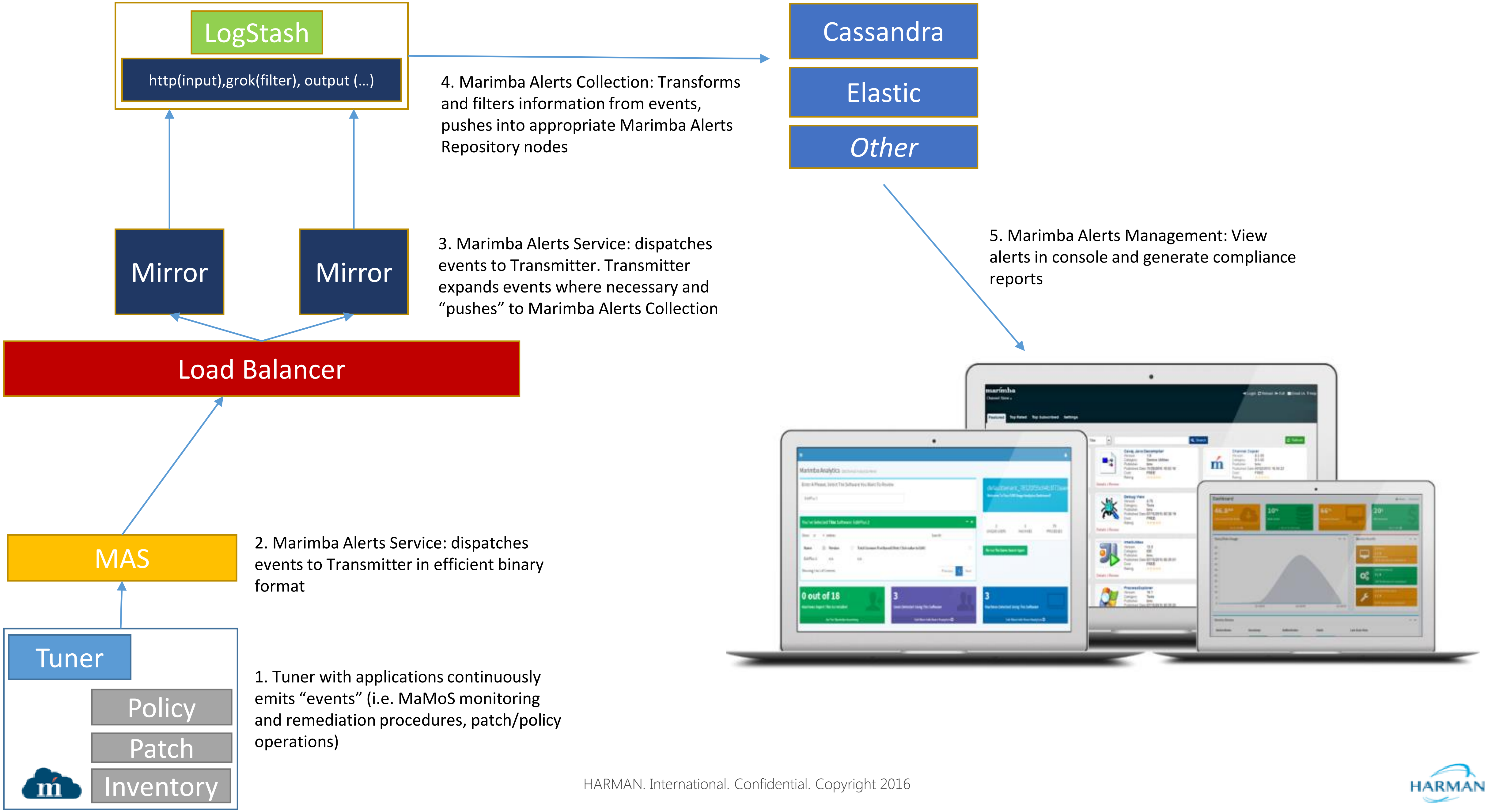
- Close to real-time status reporting into Big Data structure
    - An “add-on” channel that can work with any endpoint tuner
    - Modified Transmitter to receive status alerts
    - Integration with Big-Data store
    - Console to get real-time reports

- Implementation Details
    - Real-time reporting (effectively, event generation to event insertion is as close to real-time as possible)
    - Fast/efficient over the wire (small size of data, use integer based lookups wherever possible)
    - Customizable!
        - Add your own events for consumption
        - Support inserting of “additional” data
        - “Event capture” configuration changes driven through channel updates in Marimba
    - Utilizes the Marimba protocol
        - Existing firewall exceptions, controls, access groups, etc. just work
        - Take advantage of mirror farms for load balancing to some extent
        - Transmitter “tunnels” the report to Big Data input points
    - Configurable via properties

Architectural Components

- Tuner
  - Agent
  - Internal services (MaMoS, logging, etc.)
  - External services (Patch Service, Policy Service and other channels)
- Marimba Alerts Service (channel that encapsulates endpoint side functionality. Also allows “hooking” of additional log sources to generate events)
- Transmitter (main contact point for Marimba Alerts Service)
  - HTTP request forwarder (note: expands incoming report prior to tunneling to Data Collection Engine)
- Marimba Alerts Collection (Transmitter connects to it in order to process received reports)
  - Logstash (filter, http input plugin, grok filter, output plugins depending on Data Storage Engine like Elastic, Slack, Cassandra, etc.)
- Marimba Alerts Repository (Repository for real time data)
  - Cassandra, Elastic (and/or any other components that customer wants to integrate)
- Marimba Alerts Management (console provided by Marimba for viewing real-time alerts)
- Custom Alerts Helper (helper utility that uses regex, etc. to understand structure of custom events and update lookup files
  - References: http://www.regexplanet.com/advanced/java/index.html

Architecture



4. Marimba Alerts Collection: Transforms and filters information from events, pushes into appropriate Marimba Alerts Repository nodes

3. Marimba Alerts Service: dispatches events to Transmitter. Transmitter expands events where necessary and "pushes" to Marimba Alerts Collection

5. Marimba Alerts Management: View alerts in console and generate compliance reports

2. Marimba Alerts Service: dispatches events to Transmitter in efficient binary format

1. Tuner with applications continuously emits "events" (i.e. MaMoS monitoring and remediation procedures, patch/policy operations)

Key Benefits

Enables customers to get better and close-to-real-time reporting, with a special focus on

- Assessing deployment success/failure rates (i.e. patching, policy deployments, etc..)
- Assessing agent health (i.e. integration with MaMoS module)
- Allowing customer-specific use cases to be enabled

Customers can

- Export and Schedule reports
- Locate Past and Current activities per machine or collection
- Extend the framework to collect custom alerts from endpoints

The Framework is

- Built on big data, can support any size data
- Extremely Optimized, powerful, secure and scalable
- Customizable

Castanet Logsight – Desktop App

Demo

Thank You

Send queries to Nitish.Shrivastava@harman.com