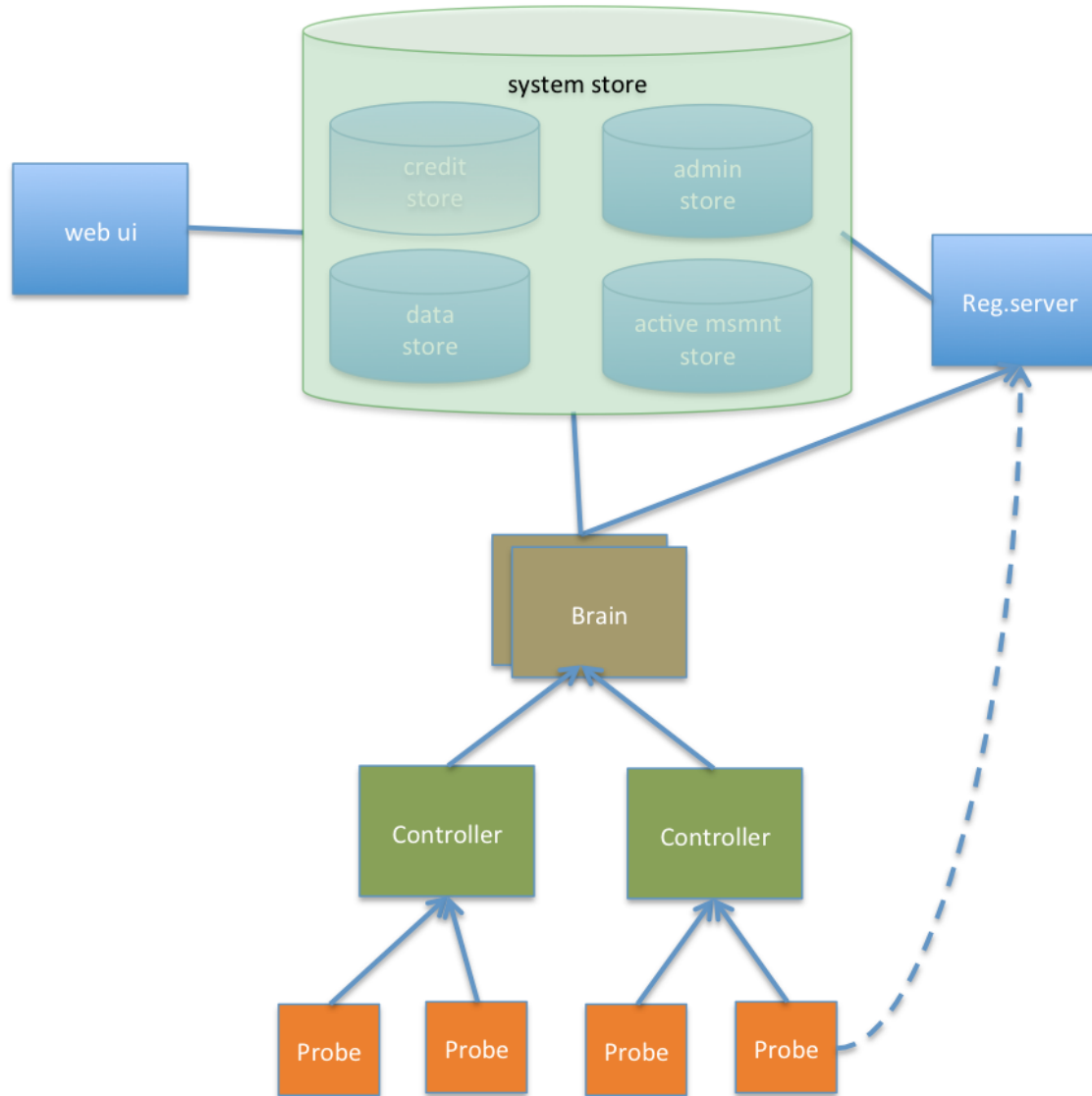


RIPE Atlas technicalities

Robert Kisteleki, RIPE NCC



RIPE Atlas - Overall Architecture

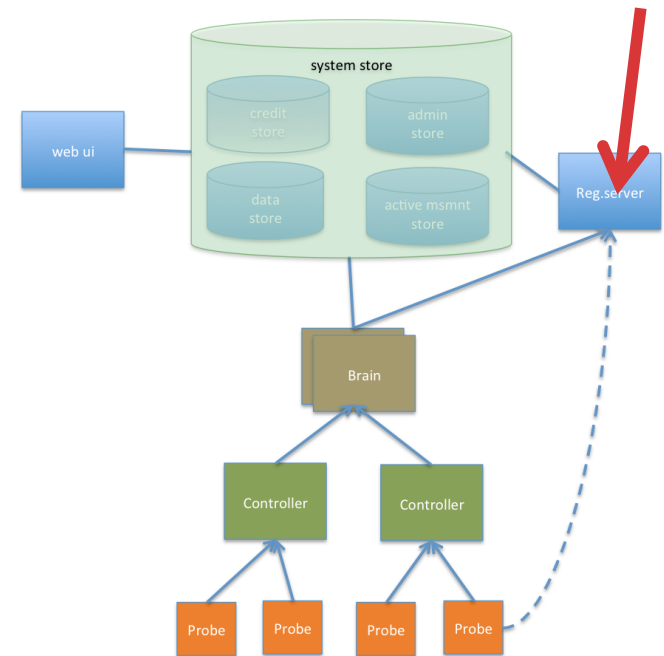


RIPE Atlas - Overall Architecture

- All components in the hierarchy maintain their connections using secure channels with mutual authentication.
- Theoretically, any component can be scaled up independently from the others
- Hierarchy allows for data aggregation
- In order to be scalable, data flow is based on “need to know”

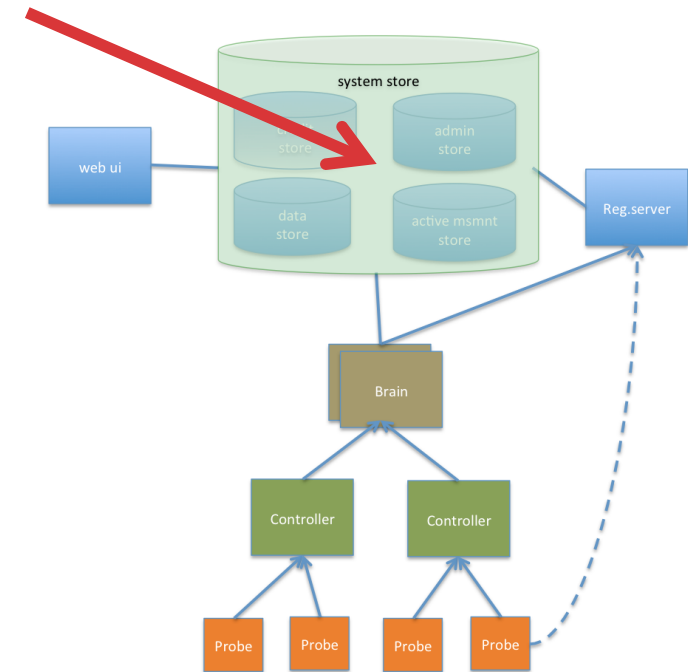
RIPE Atlas - Overall Architecture

- Registration Server:
 - The (only) trusted entry point for Probes
 - Welcomes all Probes and directs them to a suitable Controller:
 - As close as possible to the Probe
 - Not too busy
 - It has a high level overview on the current state of the system



RIPE Atlas - Overall Architecture

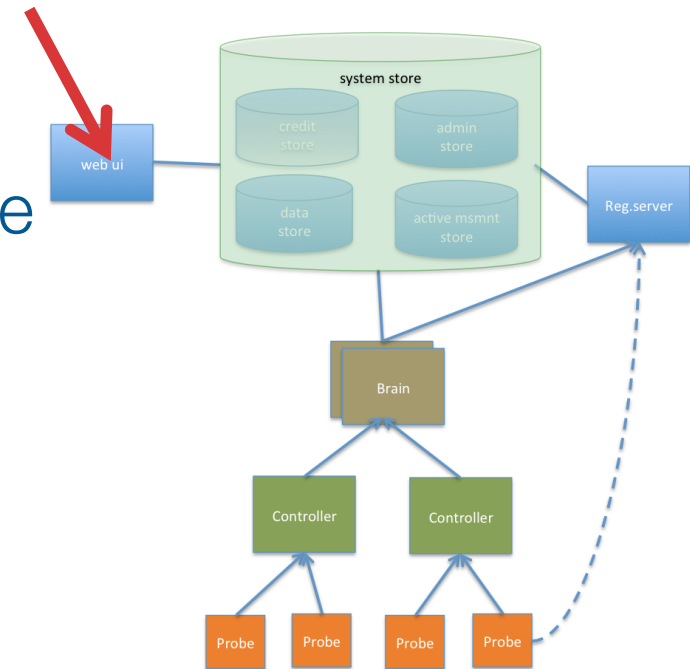
- Central database:
 - Administrative store
 - Measurement store (active store)
 - Data store
 - Credit store



RIPE Atlas - Overall Architecture

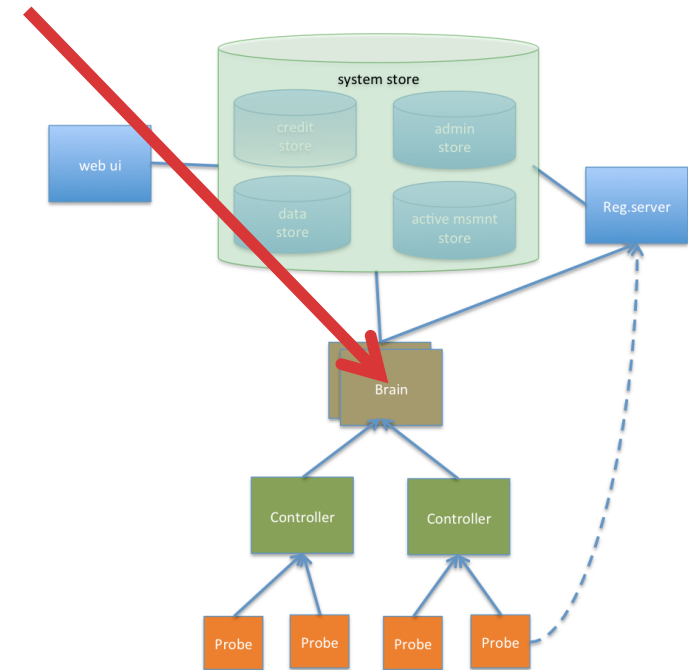
- User Interface

- Allows the users to actually use the service and look at:
 - Probe statuses
 - Measurement results
 - Community aspects



RIPE Atlas - Overall Architecture

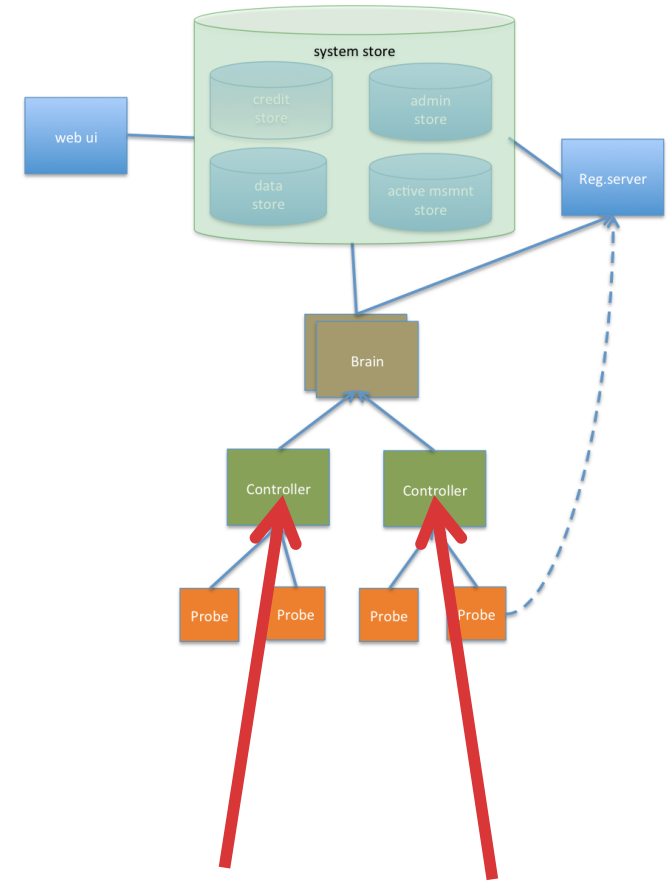
- Brain:
 - Responsible for higher order functions:
 - Coordinate measurements
 - Process ultimate results
 - Draw conclusions, maybe even act on them
 - Incorporate other sources of information, like BGP



RIPE Atlas - Overall Architecture

- Controller:

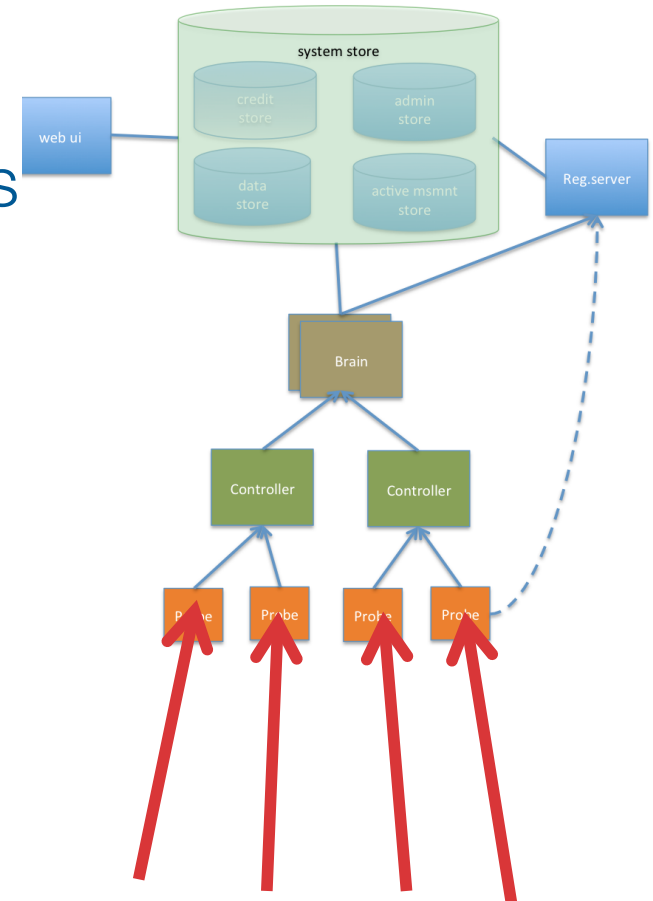
- Responsible to talk to Probes
- Assigns Probes to requested measurements based on:
 - Available Probe capacity
 - Probe locations
- Collects intermediate results and aggregates if needed
- Regularly reports to Brain



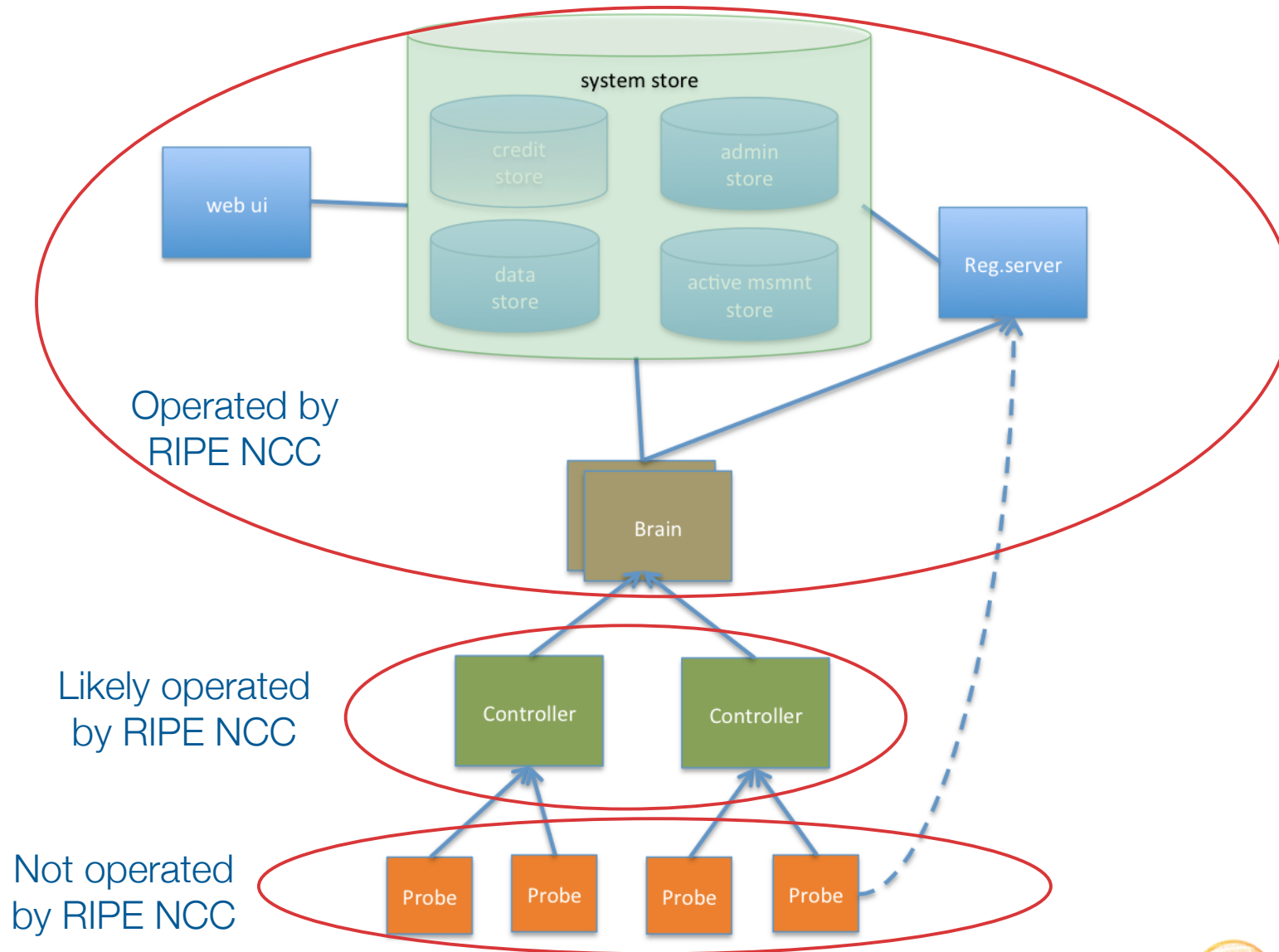
RIPE Atlas - Overall Architecture

- Probe:

- Listens to measurement commands from Controllers
- Executes built-in and dynamic measurements
- Reports results to Controller
- Other:
 - Self-upgrades if needed
 - Maintains state as much as possible



RIPE Atlas - Overall Architecture



RIPE Atlas - Probes

- Probe (v1 / generation 1):
 - Lantronix XPortPro
 - Very low power usage
 - 8MB RAM, 16MB flash
 - Runs uClinux
 - No FPU, no MMU
 - A reboot costs <15 seconds
 - An SSH connection costs ~30 seconds
 - We can remotely update the firmware



RIPE Atlas - Security aspects

- All components in the hierarchy maintain their connections using secure channels with mutual authentication.
- All information exchanges happen via channels inside a single (secure) connection.

RIPE Atlas - Security aspects

- Probes have hardwired trust material (registration server addresses / keys)
 - Upon registration, the registration server informs the probe about its future controller, and vice versa
- The probes don't have any open ports
 - They only initiate connections
 - This works fine with NATs too

RIPE Atlas - Security aspects

- Probes don't listen to local traffic, there are no passive measurements running
 - There's no snooping around
- We suspect we'll lose some probes because of “deep interest in how they really work”. That is:
 - Some will be disassembled
 - Some will be hacked locally, modified and used for something else
 - But there is no shared key material on the Probes...

RIPE Atlas - Other Bits and Pieces

- IPv6 support:
 - The system in general supports IPv6
 - We already do IPv6 measurements
 - However, one software component is “unfinished” yet, so we must connect to the controllers over IPv4
 - Therefore, as of now, IPv4 is needed for Probe operation
 - We’ll work together with the vendor to fix this

RIPE Atlas - Other Bits and Pieces

- The Probe has no direct user interface to configure anything on it
 - So DHCP is a must for IPv4, RA is needed for IPv6
 - Deployment in places without DHCP is not yet supported
 - But we do have ideas on how to solve this

Questions?

