SECURING AGAINST INTRUDERS AND OTHER THREATS
THROUGH A NFV-ENABLED ENVIRONMENT
[H2020 - Grant Agreement No. 700199]

The role of Trusted Computing in the SHIELD project

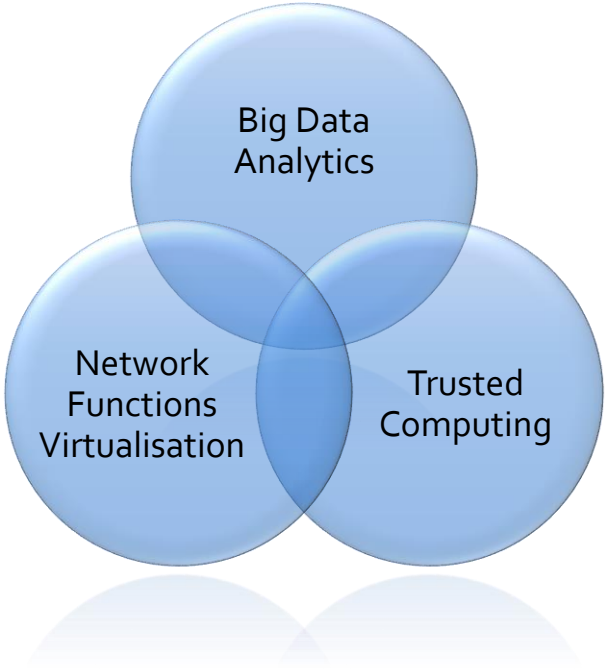
Ludovic Jacquin, Hewlett Packard Enterprise.

TCG members meeting
Lisbon, Portugal, 16 October 2018

Overview of SHIELD

SHIELD's mission & concept

SHIELD aims to deliver an open solution for dynamically establishing and deploying virtual Security infrastructures in ISPs and corporate networks.



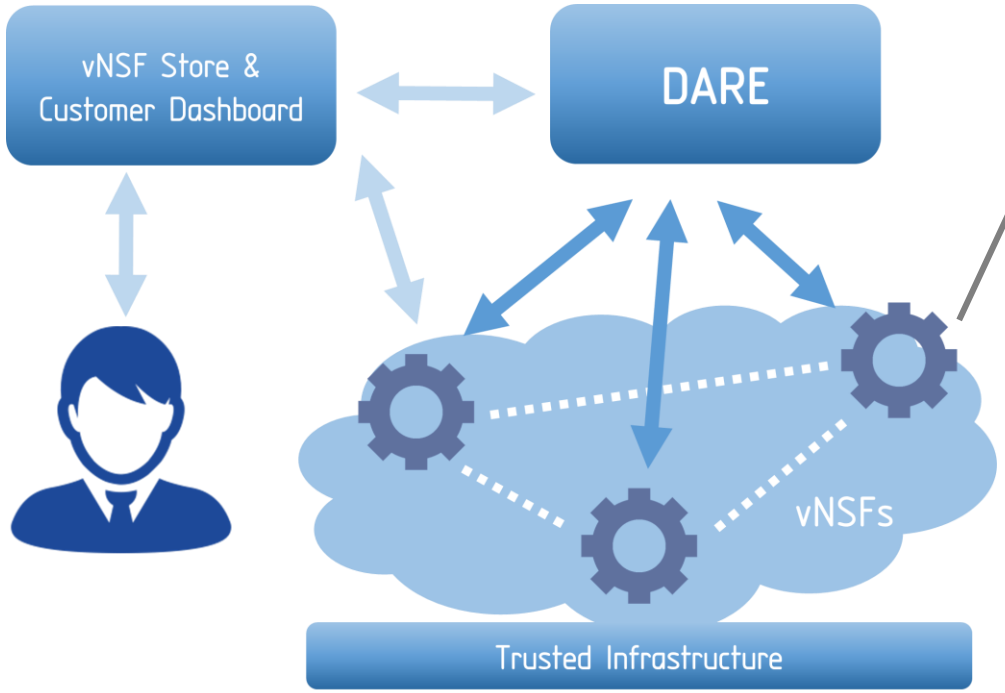
SHIELD in a nutshell

- Virtualised security functions gather security monitoring events
  - The events and metrics are store in a big-data store
- Security-oriented machine learning algorithms detects and predicts vulnerabilities and attacks
- Remediation of the attack through (partially) automated deployment of enforcing security functions
  - An operator is required to approve the suggested remediation

The virtualisation and data/control separation gaps

- Virtual security functions alone cannot be verified, evaluated.
  - The state is remotely controlled by the orchestration.
- How to verify if a virtual function is working as intended?
  - Securely
  - Automatically
- Also, the network is dynamic – use of SDN paradigm.
- Trusted Computing technologies and mechanisms!

The SHIELD system components (I)



VIRTUAL NETWORK SECURITY FUNCTIONS (VNSFS)

SHIELD offers Security as-a-Service (SecaaS) based on virtualised Network Security Functions (vNSFs).

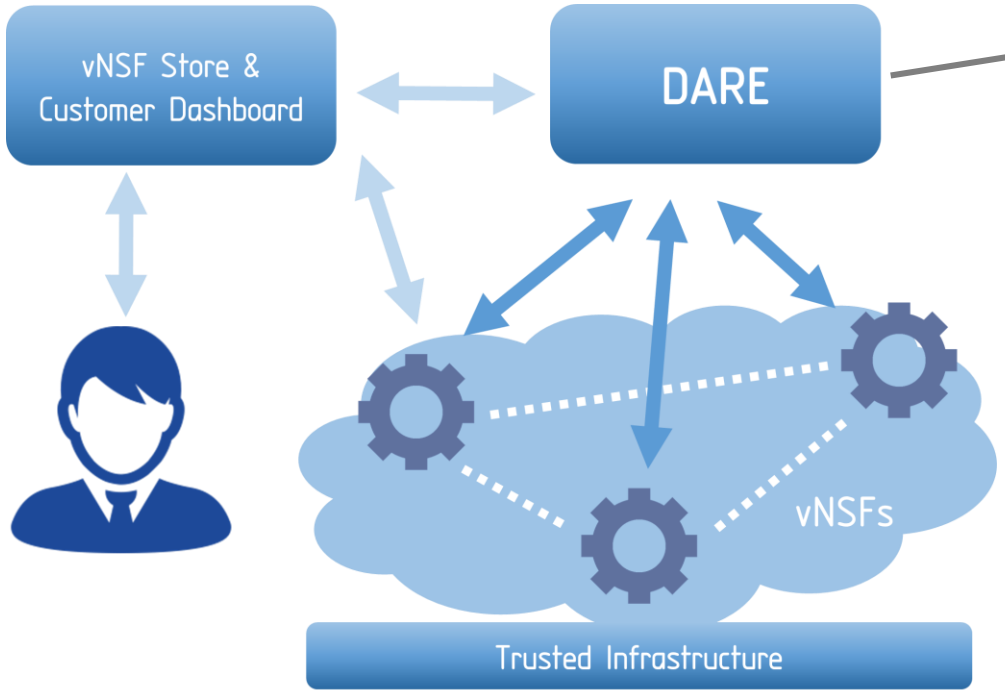
vNSFs are instantiated within the network infrastructure by a vNSF orchestrator in order to effectively monitor and filter network traffic in a distributed manner.

Advertisement, browsing, selection and trading of vNSFs in a secure manner is provided by a logically centralised repository (Service Catalogue)

KEY TECHNOLOGIES

openstack. OPEN DAYLIGHT Open Source MANO TNOVA sonata

The SHIELD system components (II)



KEY TECHNOLOGIES

hadoop

APACHE Spark™

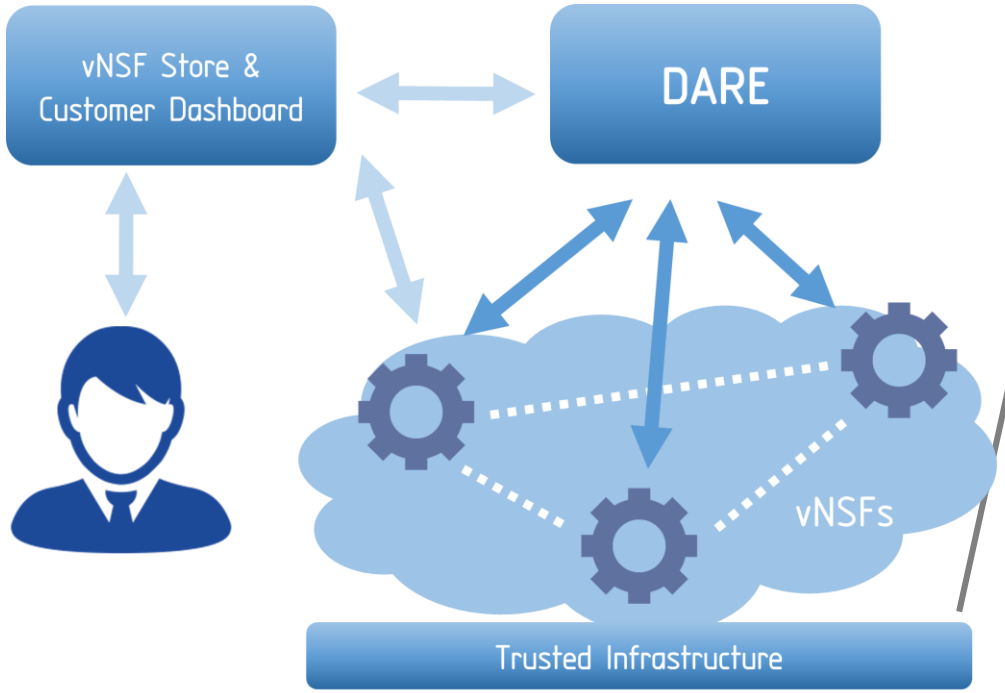
Apache Spot

DATA ANALYSIS AND REMEDIATION ENGINE (DARE)

DARE is an information-driven IDPS platform capable of predicting specific vulnerabilities and attacks by relying on Big Data, Threat Monitoring and Machine Learning to analyse the output produced by vNSFs.

Pattern discovery techniques analyse data to identify current malicious behaviours or predict likely threats. Analysis' results are accessible by systems and security administrators via a dashboard.

The SHIELD system components (III)



TRUSTED INFRASTRUCTURE

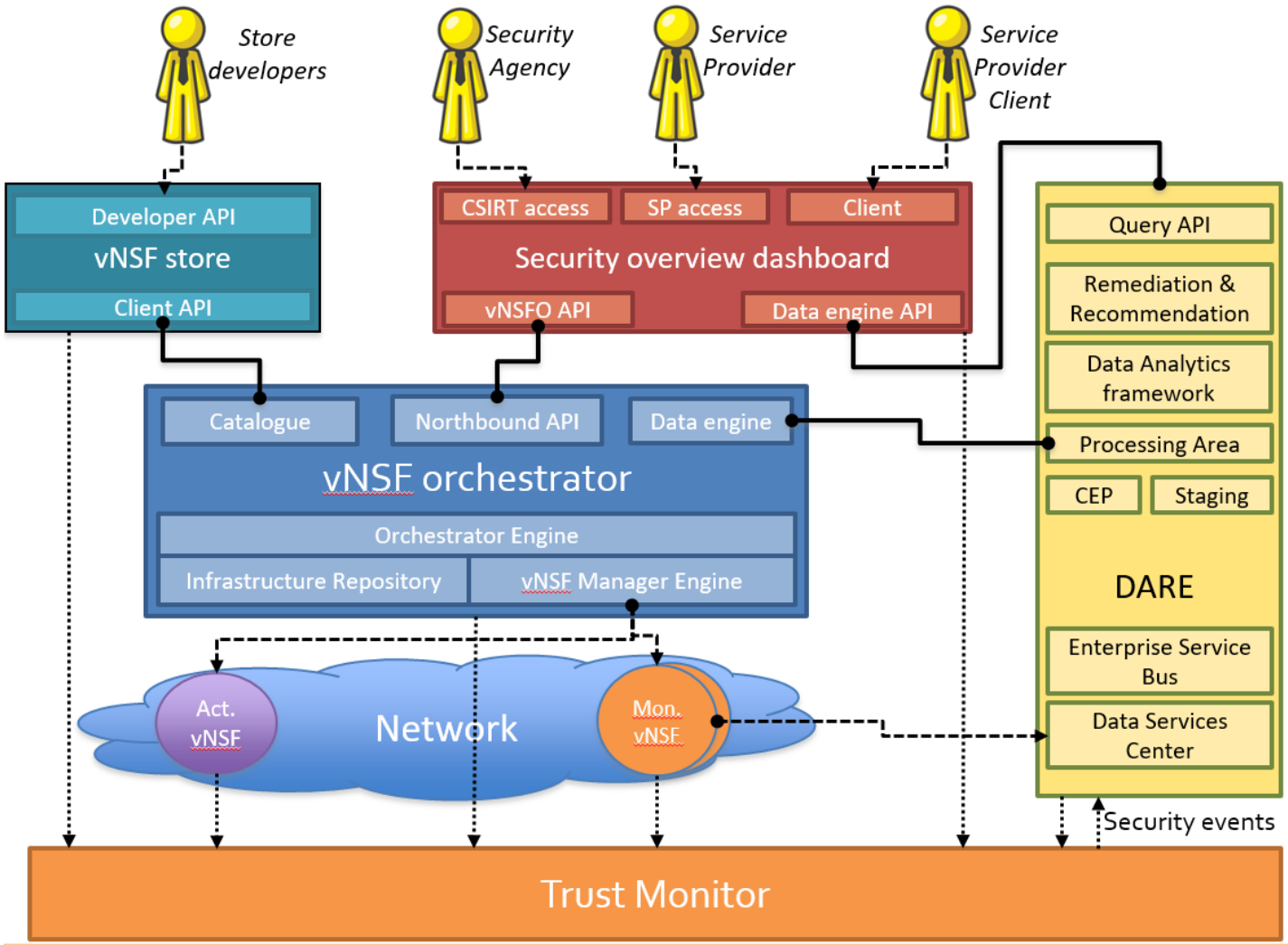
The trustworthiness of the secure SHIELD framework is implemented by relying on Trusted Computing technologies. The infrastructure attestation binds the vNSFs and the network configuration with the store and orchestration of the network.

The key components of the secure SHIELD framework are protected using Trusted Platform Modules (TPM), assuring the integrity of the software and the configuration.

KEY TECHNOLOGIES

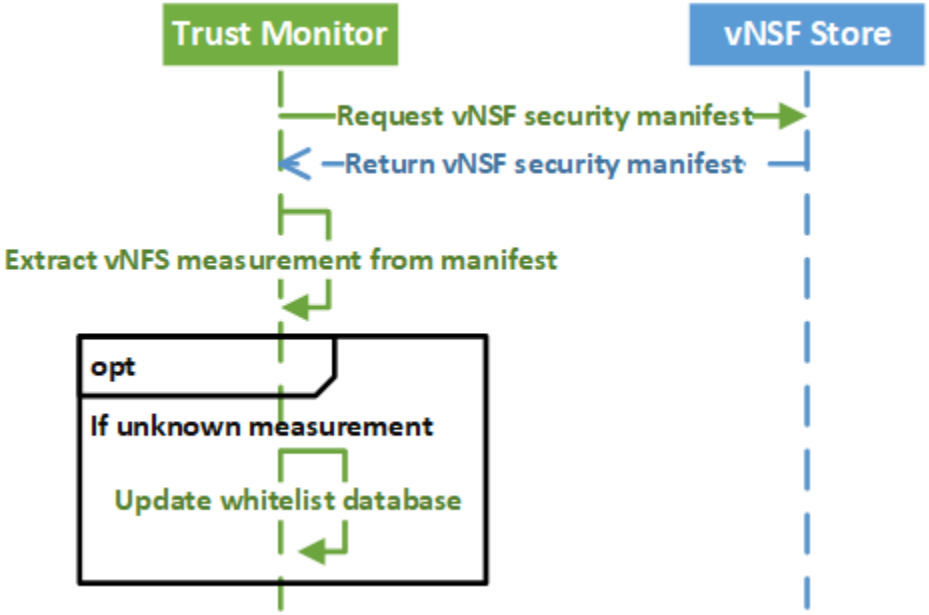
SHIELD design and architecture

SHIELD architecture



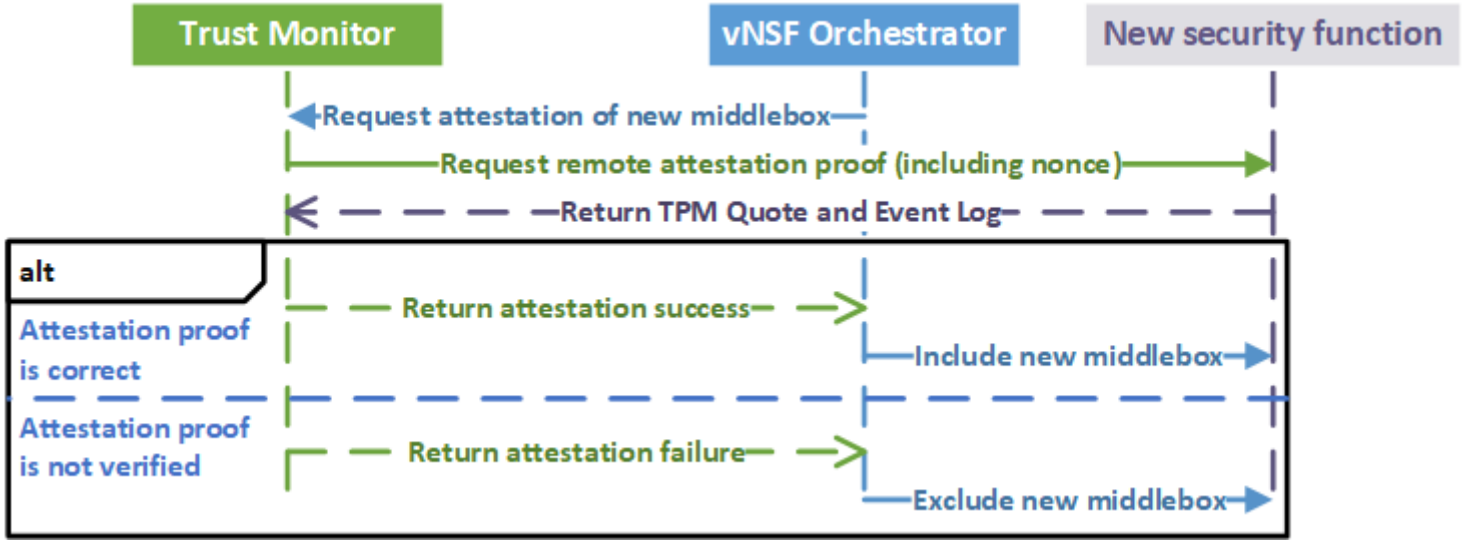
Golden value for virtual security functions

- Golden value created by the developers
  - Part of the security manifest



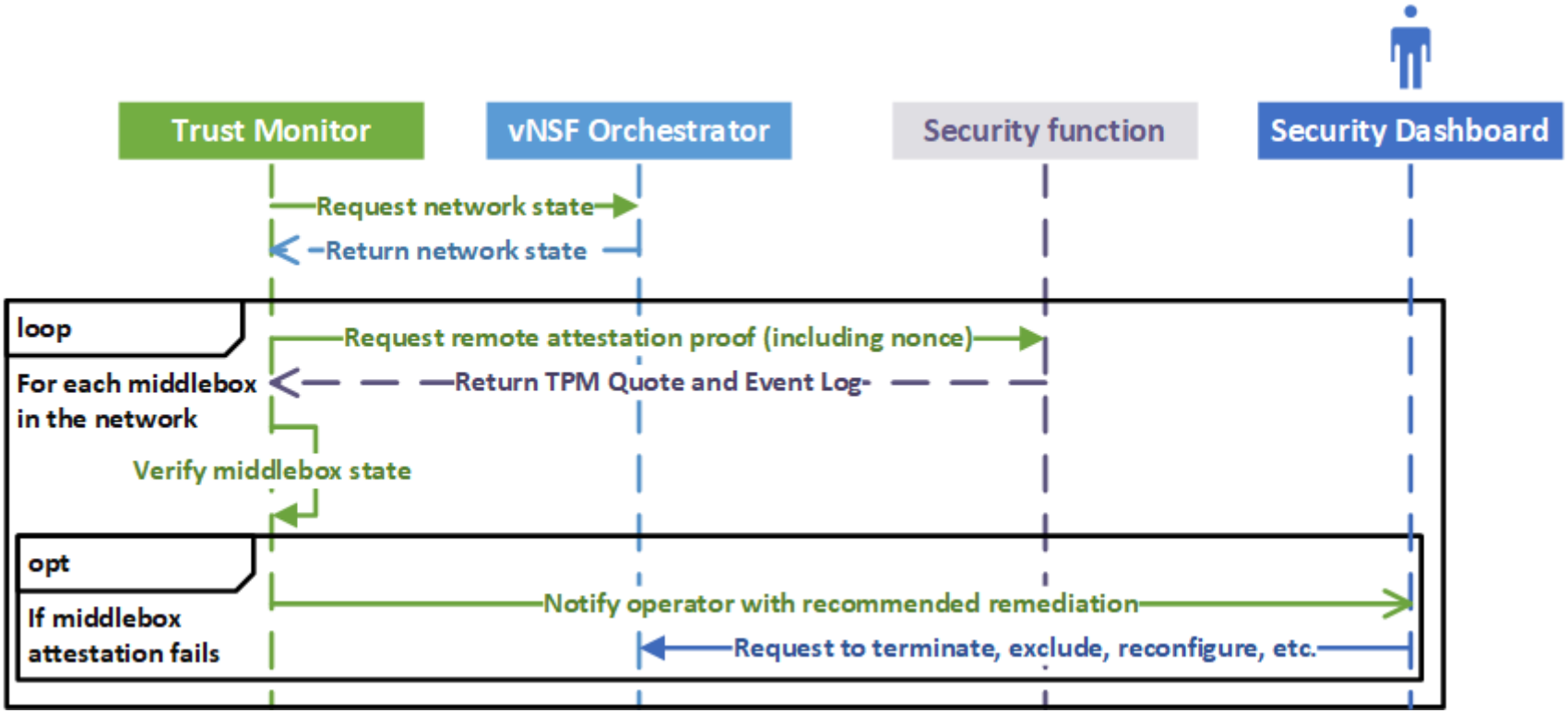
Initial deployment of a security function

- Security functions (and physical nodes) are verified before use



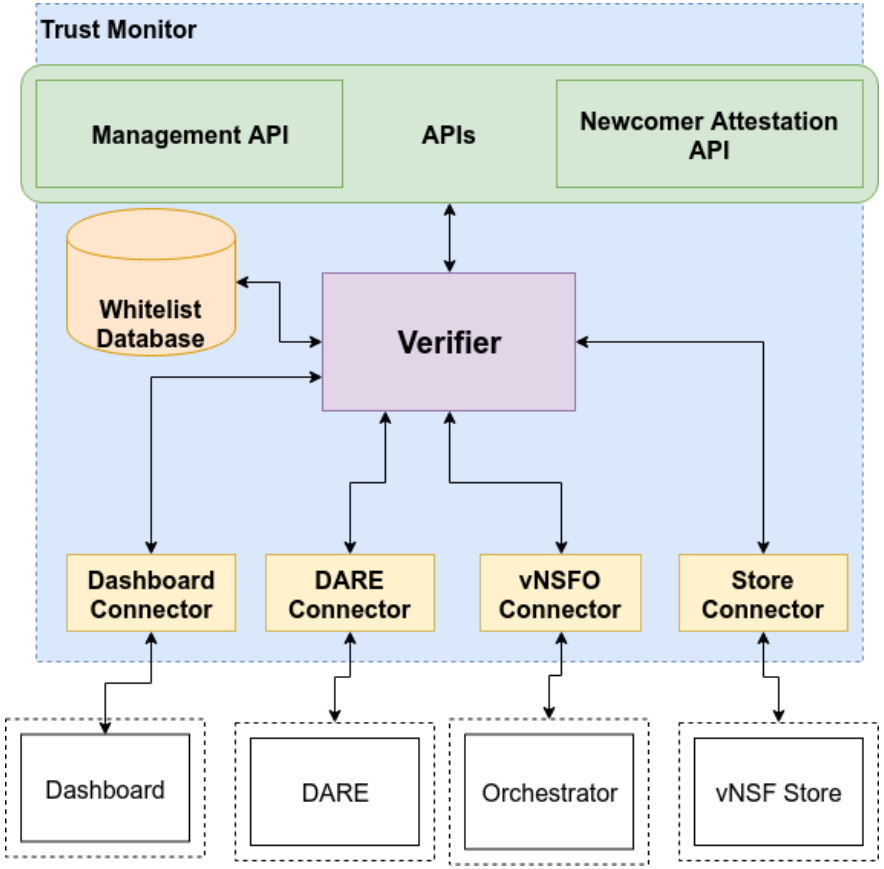
Periodic attestation of security functions

- Continual attestation through the security functions lifecycle



Trust Monitor detailed architecture

- History of attestation logs are stored in the DARE (for audit)



Lesson learnt on implementing attestation (Trust Monitor)

Servers and middleboxes attestation (1/2)

- Boot-time integrity verification of the servers firmware
  - Via Measured Boot for TPM 1.2
- Run-time integrity verification of binaries and configuration files in servers and middleboxes
  - Based on the Linux *Integrity Measurement Architecture* (IMA) kernel module + custom Linux kernel
  - Tailored for the Docker lightweight virtualisation environment
- Automated generation/update of whitelist database
  - Including reference measurements from software repositories (limited to CentOS Linux distribution)

Servers and middleboxes attestation (2/2)

- Takes around 5 to 6 seconds RTT w/ verification via whitelist database (up to 128 active containers)
  - Hard limit caused by the TPM signature over the integrity report (around 2 seconds) + generation time of the integrity report
  - The verification process has a smaller influence on the overall time
- The total average time grows linearly with the number of active containers
- Gap: missing integration with NFV Virtual Infrastructure Manager

SDN switches attestation

- Detection of rogue SDN controller
- Detection of incorrect SDN rules

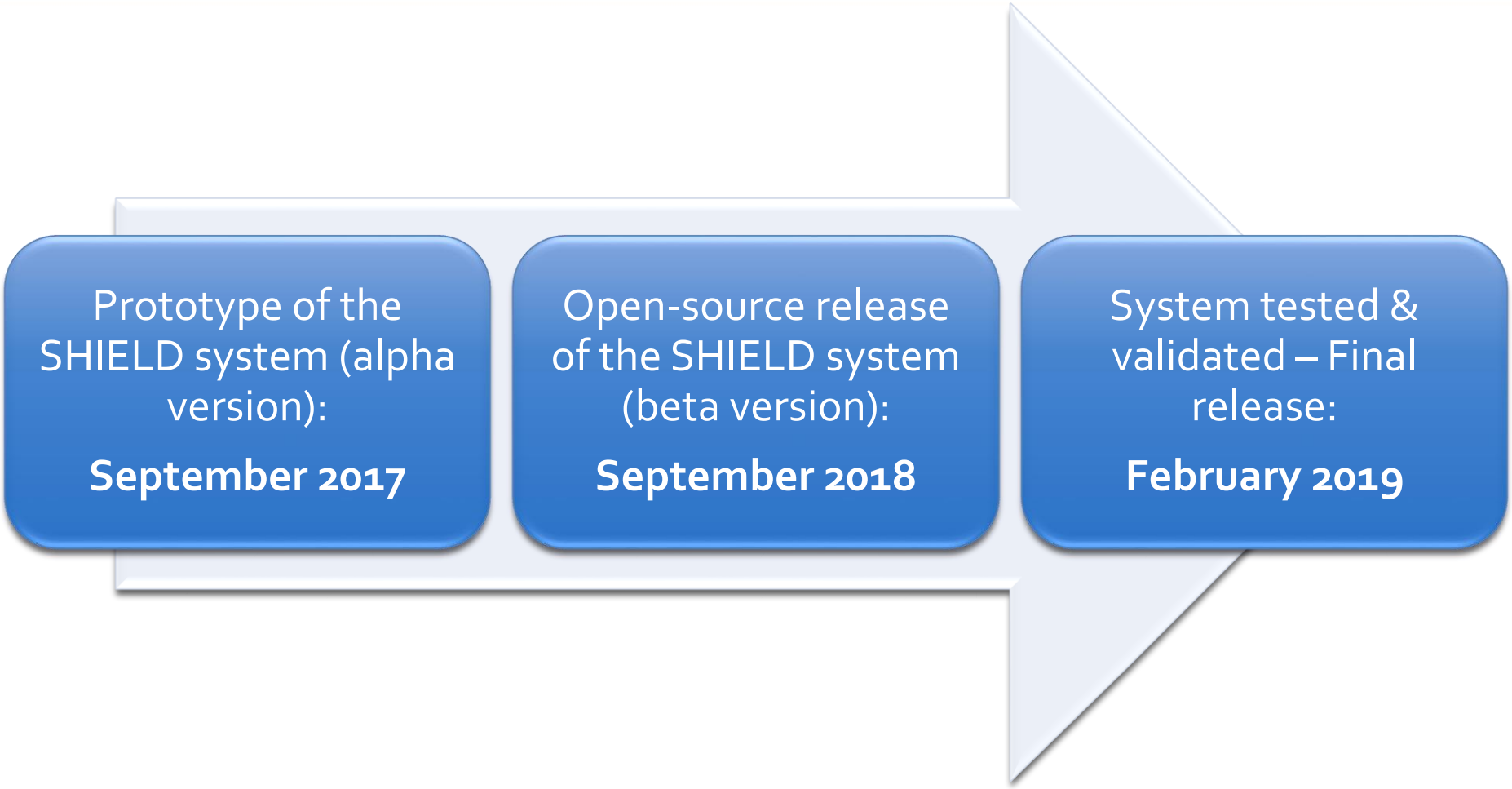
- Can handle 10k OpenFlow rules
  - Bottleneck between SDN controller & Trust Monitor
- Gap: no common (SDN controller-switch) identifier for rules

- Around 2 to 3 seconds RTT
  - 600ms for the TPM signature itself

On-going effort

- Include recommended remediation
  - Exclude node, update configuration, reconfigure OpenFlow rules
  - 1-click complete automation
- Whitelist nodes allowed to communicate with vNSFO
  - Based on being verified by the Trust Monitor
- Integration with the other components

Key project milestones



Follow us!

https://www.shield-h2020.eu/

@shield_h2020

SHIELD EU Project

info@shield-h2020.eu

https://www.youtube.com/channel/UCXBxrz-5eReK4nSC46yks5A

SHIELD has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 700199