Security-Mode ONOS

Seungwon Shin
Changhoon Yoon
KAIST

Philip Porras
Martin Fong
SRI International

Thomas Vachuska
Brian O'Connor
ON.LAB

ON.LAB

SRI International KAIST

Security-Mode ONOS ?

1) Application layer access control

- A network application permission-enforce model for managing distributed ONOS applications

2) Application auditing

- Provide operators with explicit insight and control over the ONOS core services and APIs used by each ONOS application

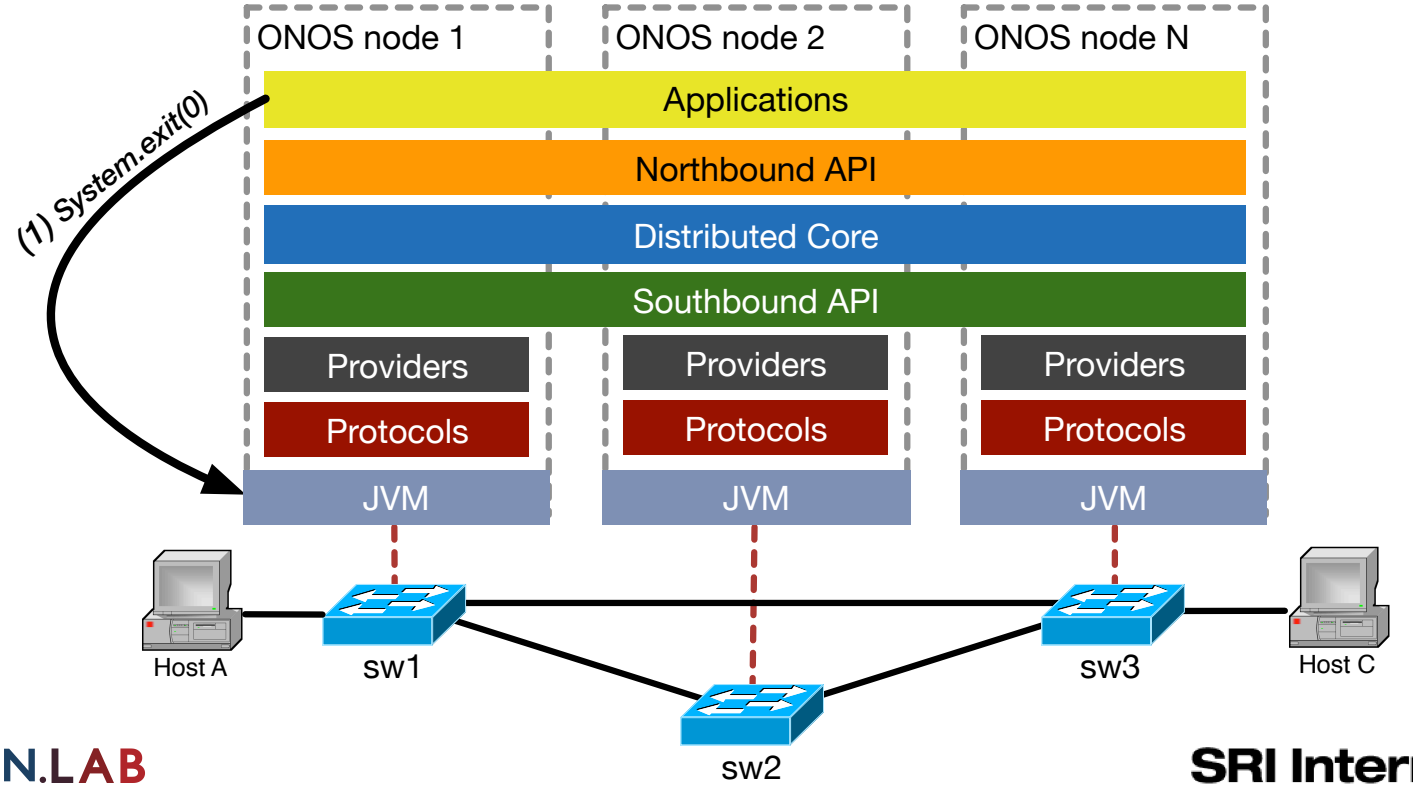
Motivation

- **Powerful authority granted to ONOS applications**
  - Direct network manipulation
  - Data exfiltration
  - Denial of service
  - anything is possible!

- **Applications cannot be trusted**
  - Third-party applications (from untrusted source) may contain malicious/buggy code

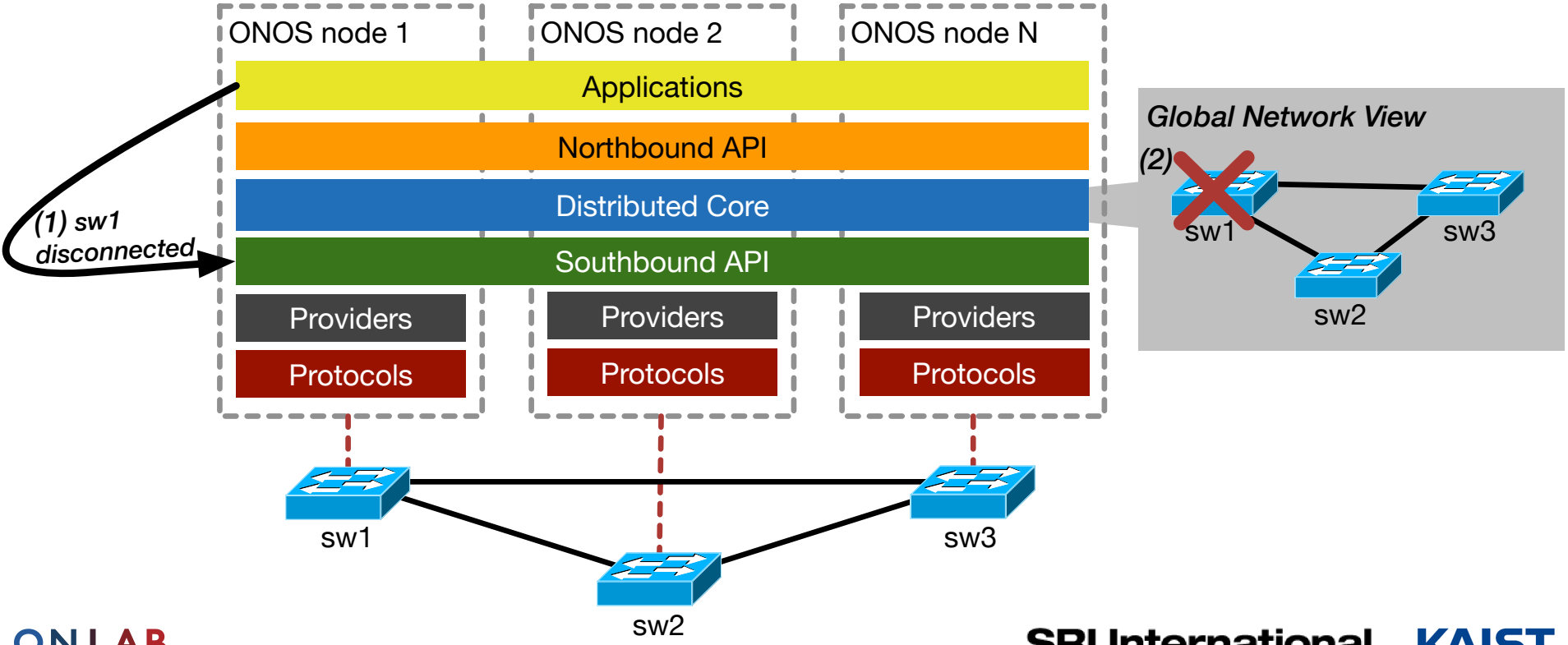
Motivating examples

- System command execution – Denial of Service



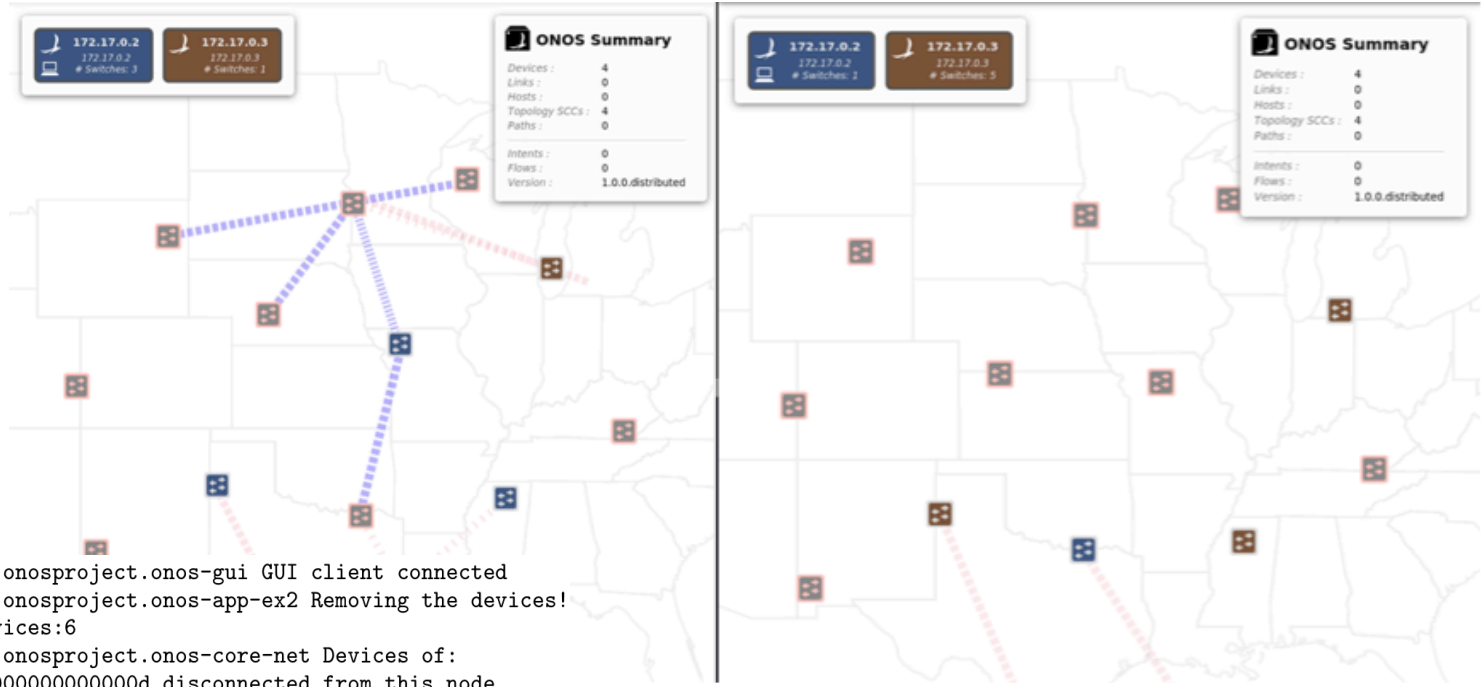
Motivating examples

- Southbound API access



Motivating examples

- Persistent switch disconnection event generation

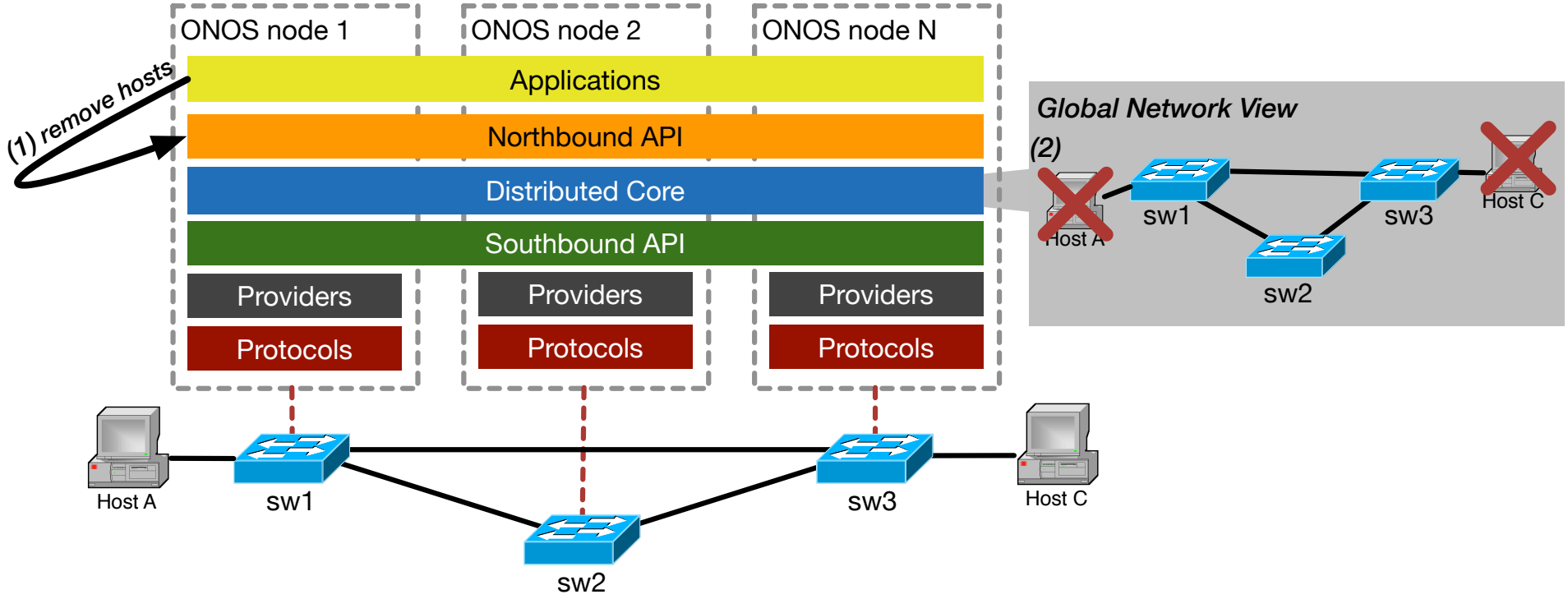



```
| 117 - org.onosproject.onos-gui GUI client connected
| 160 - org.onosproject.onos-app-ex2 Removing the devices!
        #devices:6
| 145 - org.onosproject.onos-core-net Devices of:
        000000000000000d disconnected from this node
```

Inconsistent global network view

Motivating examples

- Administrative API access



Motivating examples

- Host and device removal

Permission Model

1) Bundle-level role-based access control

- If a bundle is an application bundle,
  only grant **minimum** permissions required
  (e.g., system command execution, Southbound API access are restricted)

2) Application-level role-based access control

- If application has ‘USER’ role,
  Administrative NB API access is restricted

3) API-level permission-based access control

- An application can access NB API,
  ONLY IF it has the required permission

Security Policy File

```
1  <security>
2      <role>USER</role>
3      <permissions>
4          <app-perm>DEVICE_READ</app-perm>
5          <app-perm>TOPOLOGY_READ</app-perm>
6          <app-perm>FLOWRULE_WRITE</app-perm>
7          <osgi-perm>
8              <classname>ServicePermission</classname>
9              <name>org.onosproject.demo.DemoAPI</name>
10             <actions>get,register</actions>
11         </osgi-perm>
12         <java-perm>
13             <classname>RuntimePermission</classname>
14             <name>modifyThread</name>
15         </java-perm>
16     </permissions>
17 </security>
```

Application role

Application permissions

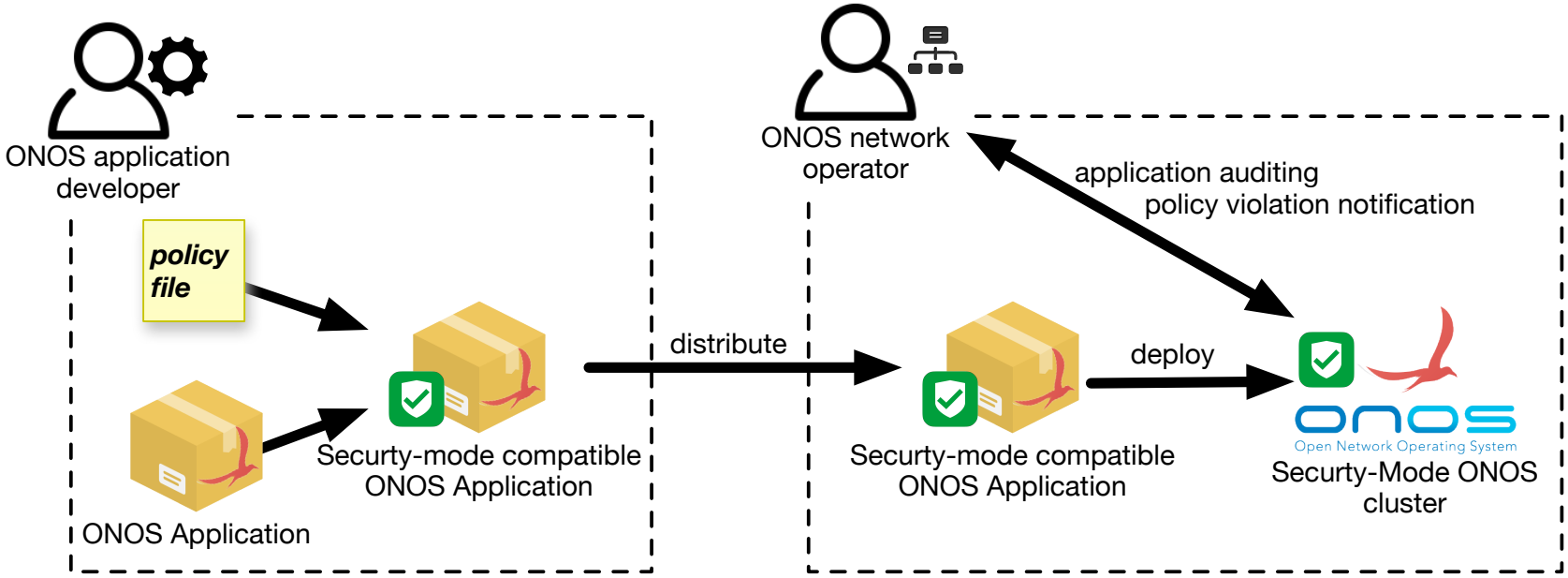
Extra OSGi permissions

Extra Java system permissions

Provides a clear view of application intrinsic

Security-Mode ONOS (SM-ONOS)

- The BIG picture



Secure application activation

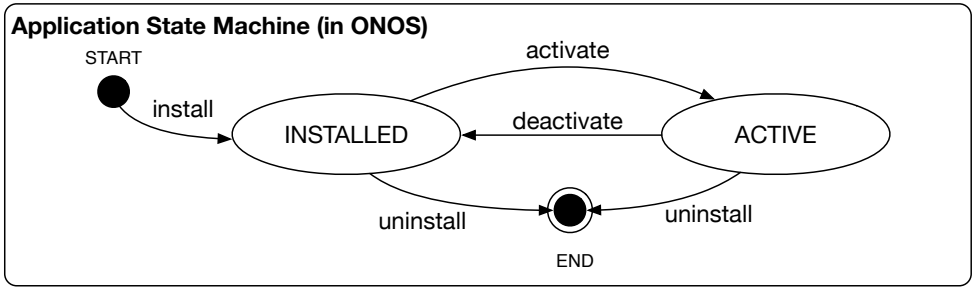
- In **Security-Mode**, an ONOS operator MUST…
  1) review application security policy
  2) agree and accept security policy
  … prior to activating an application

- Upon the policy acceptance, policy is *immediately* enforced to the application bundles

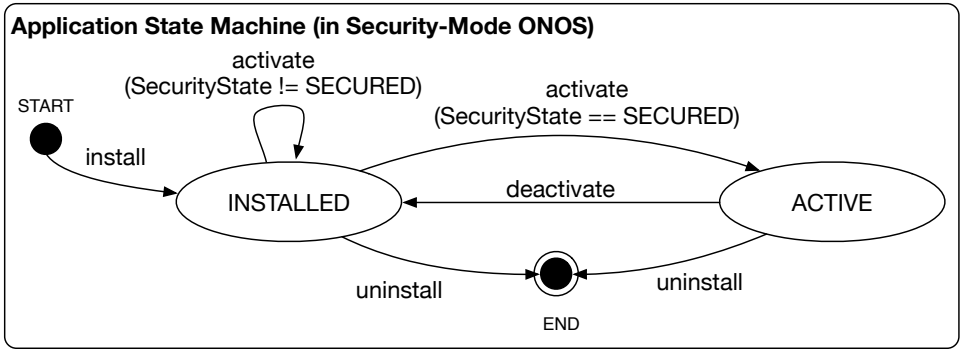
- ONOS operator should be able to go through the secure app activation process ANYWHERE in the cluster!

Secure application activation

- Extend existing application state machine

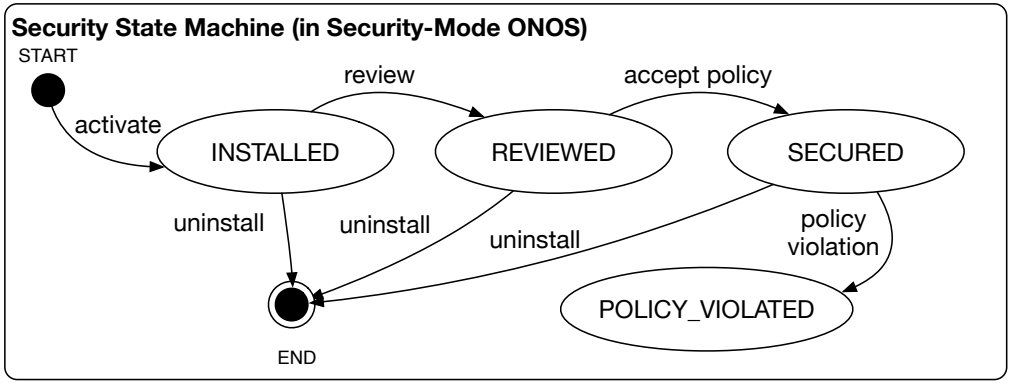


- An application can only be activated, if '**Security State**' of the application is '**SECURED**'



Security State of Application

- SM-ONOS maintains Security State of each application
  (via Strongly consistent / RAFT-based consistent store)
- **Security States**
  - **INSTALLED** : Application has been installed
  - **REVIEWED** : Security policy has been reviewed by ONOS operator
  - **SECURED** : Security policy has been accepted and enforced
  - **POLICY_VIOLATED** : Application has violated security policy



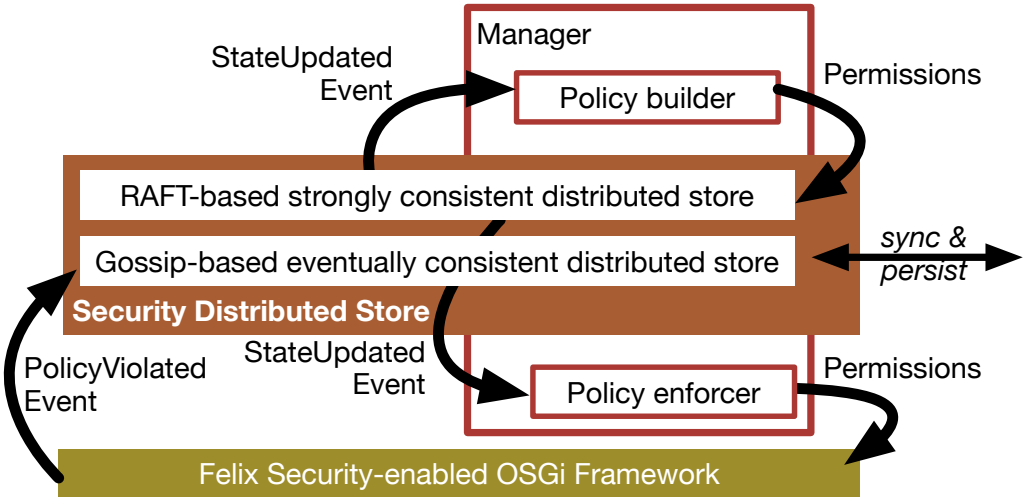
Distributed security policy enforcement

- Application subsystem enables
  'Cluster-wide' application installation/activation

- Application security policy **MUST** be **consistently** enforced to distributed applications!

- **How?**
  - Consistent distributed store

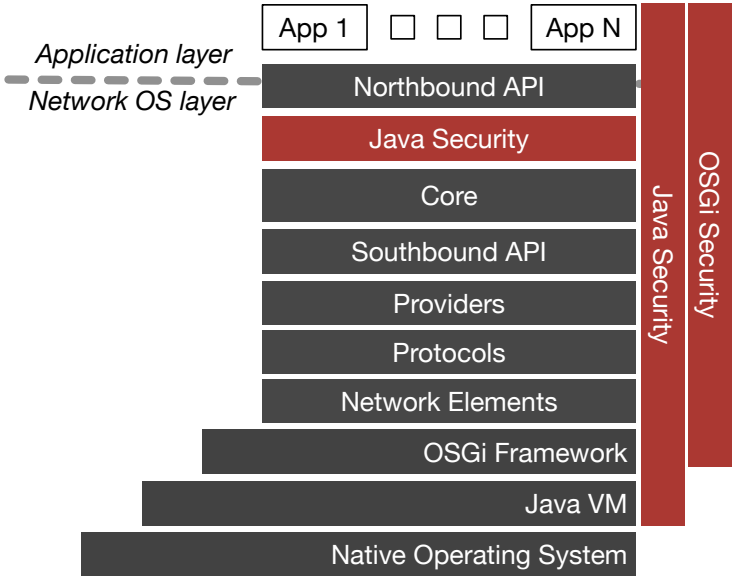
Distributed security policy enforcement

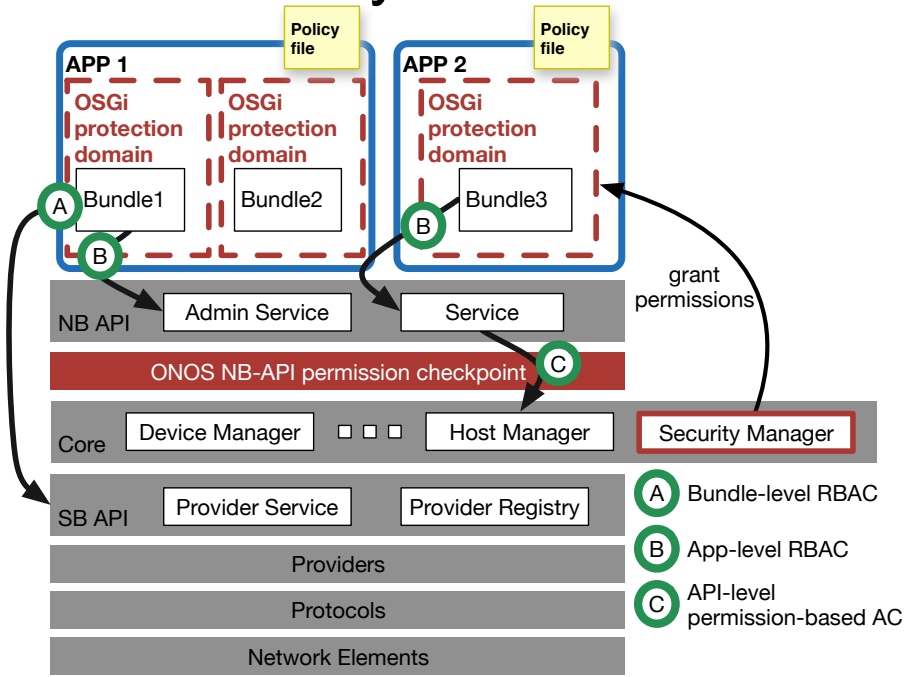


- **Security policy and state information**
  - synchronized using **strongly** consistent store
- **Security policy violation events**
  - synchronized using **eventually** consistent store

Application security policy enforcement

- **How to locally enforce security policy ?**
  - Leverage both Java and OSGi security framework





Summary

- New application layer access control
- Secure application activation mechanism
- Distributed application security policy enforcement

Thank You

Learn more about ONOS and join the community at
onosproject.org

"Software-defined networking can radically reshape the wide area network. The introduction of **ONOS** provides another open source SDN option designed for service provider networks with the potential to deliver the performance, scale, availability and core features that we value"

John Donovan
Senior Executive Vice President
AT&T Technology & Operations

at&t

BUILD

USE

CHAMPION

ON.LAB

SRI International

KAIST