integrated
architectures

The Smart Grid
Security Innovation Alliance

Security Fabric

John Reynolds

October 26, 2011

Cambridge, Massachusetts

The SGSIA addresses the entire ecosystem.

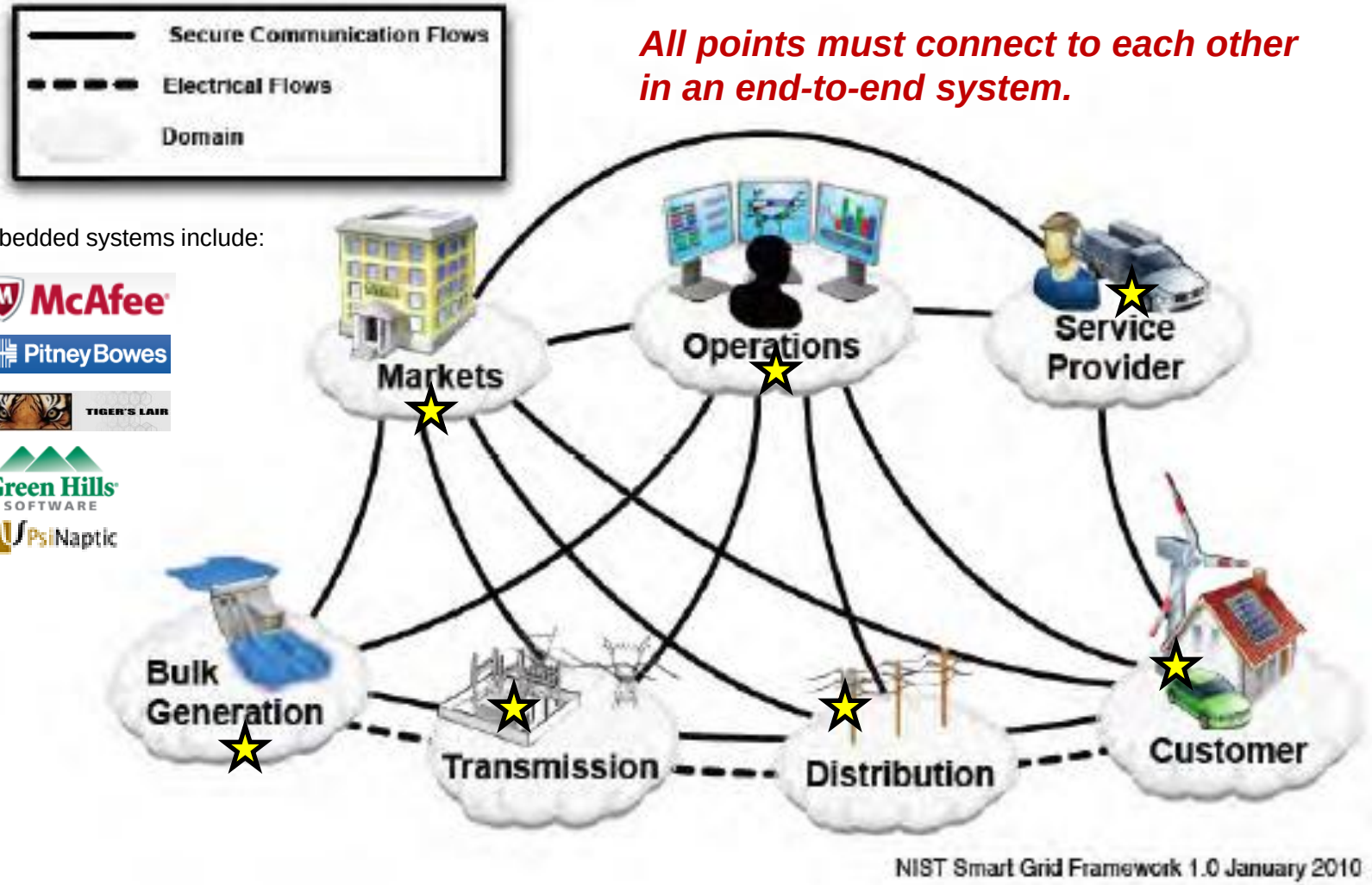
- The Smart Grid Security Innovation Alliance is a working association dedicated to practical deployment of the smart grid complex system solution in the United States:
  - Utilities
  - Systems integrators
  - Manufacturers
  - Technology partners
  - National certification and interoperability entity
- The alliance is intended to give the CEO of a utility the purview of up-to-the moment knowledge of the options available to make wise investment decisions regarding infrastructure deployment for optimal returns.

The variation includes the proper orientation for large, medium, and small utilities.

SGSIA Participants

- **First Build**
  - Integrated Architectures
  - Drummond Group
  - Sypris
  - SAIC
  - HereNow
  - TeamF1
  - Tibco
  - NitroSecurity
  - Pitney Bowes
  - McAfee
  - Tiger's Lair
  - PsiNaptic
  - Green Hills
  - DTI
- **Subsequent Builds**
  - Verizon
  - Schweitzer Engineering Labs
  - M2M Dynamics*
  - Coulomb
  - Wurldtech
  - OSIsoft
  - SNMP Research
  - Honeywell
  - Wind River
  - VeriSign
  - Entrust
  - SafeNet
  - Thales
  - Microsoft
  - Nakina
  - Telcordia
  - Itron
  - Echelon
  - Ambient
  - Mocana

Our strategy is to provide certified interoperability to the key devices controlling the grid.



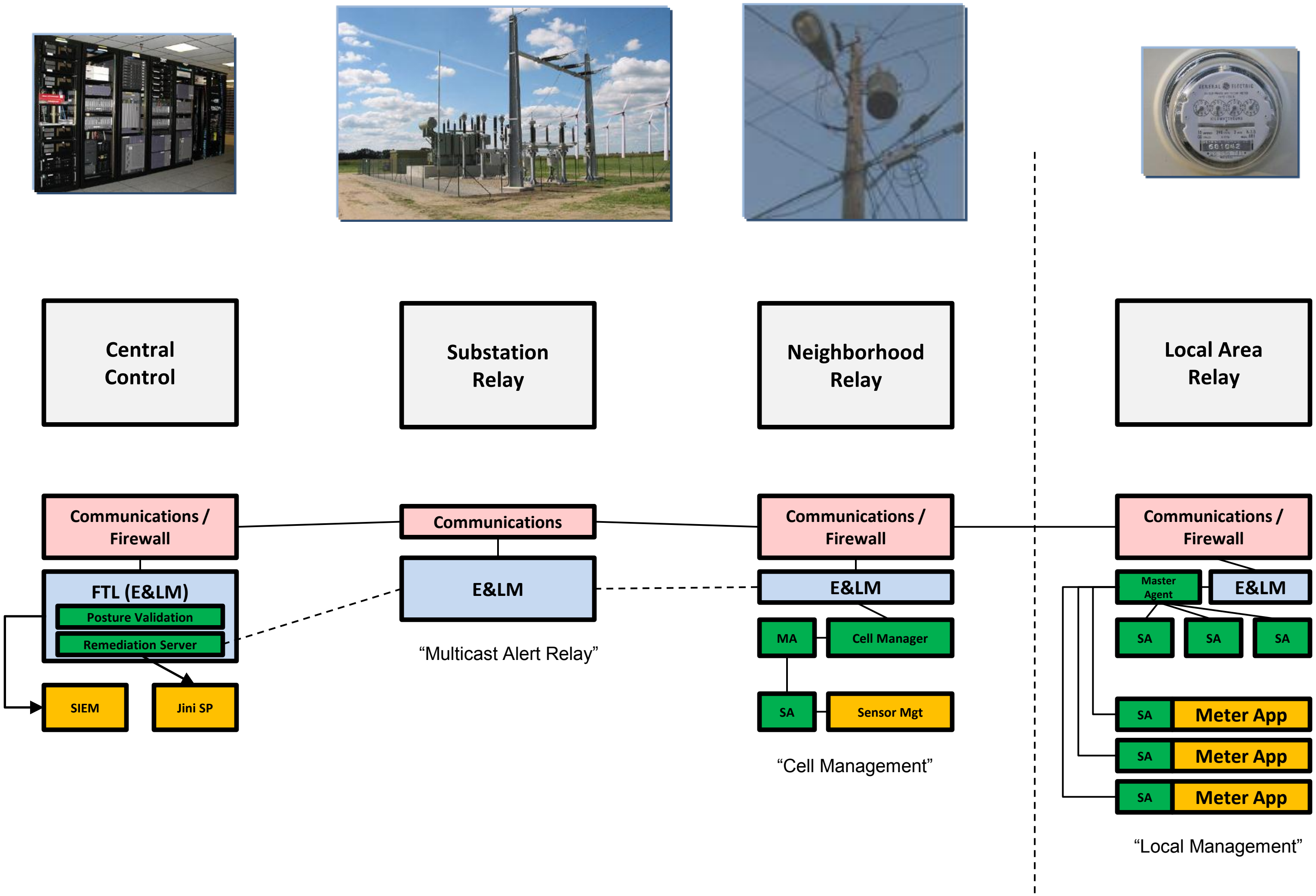
All points must connect to each other in an end-to-end system.

The embedded systems include:

- McAfee
- Pitney Bowes
- Tiger's Lair
- Green Hills Software
- PsiNaptic

The HSM solution would be embedded at each critical point in the energy infrastructure.

The general approach to power distribution requires a thin overlay of end-to-end management services.



The Department of Energy
Tailored Trustworthy Spaces

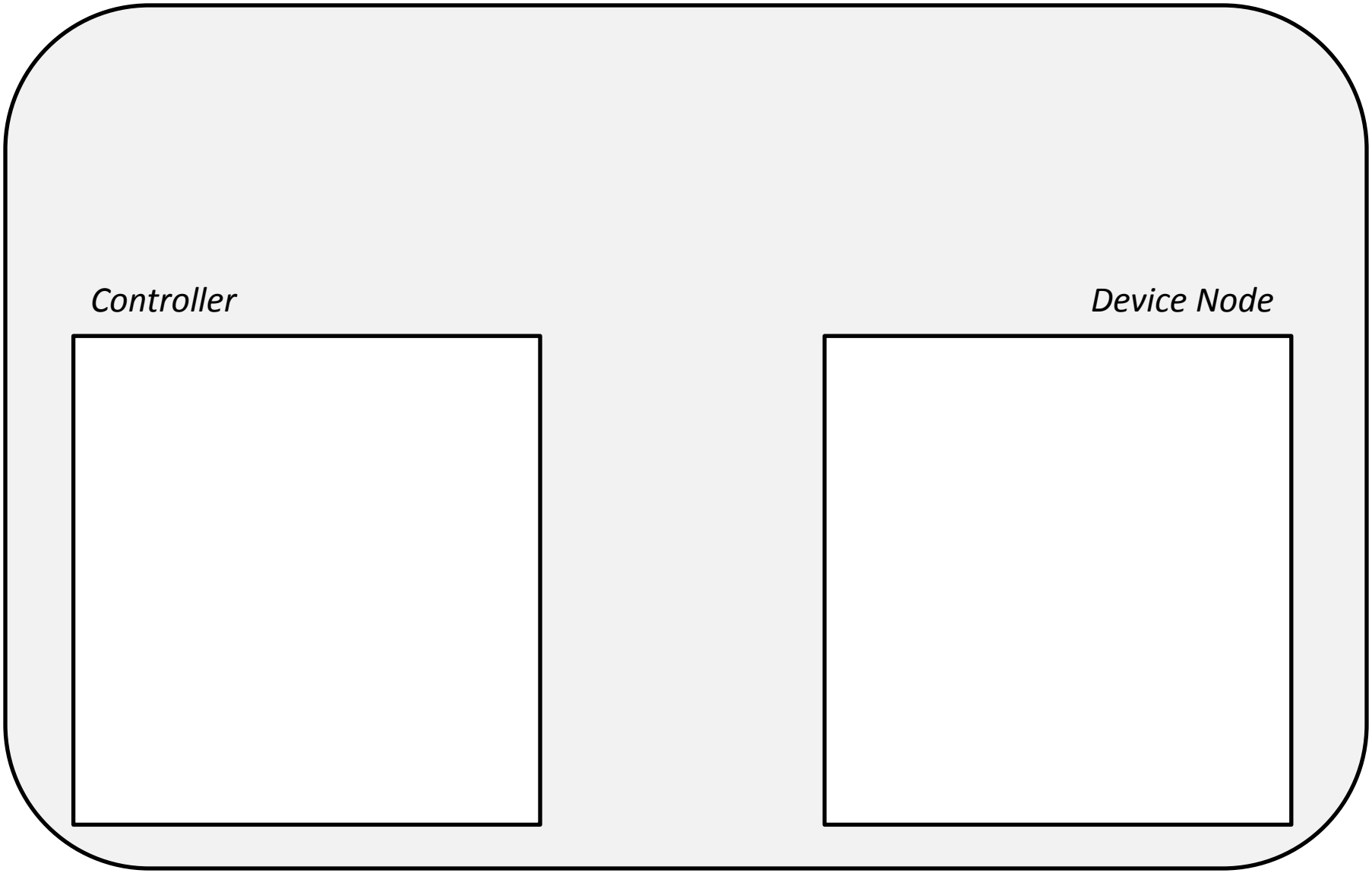
A tailored trustworthy space (TTS) provides a flexible, adaptive, distributed operating system environment for a set of devices and applications that can support functional and policy requirements arising from a wide spectrum of activities in the face of an evolving range of threats.

A TTS recognizes a device's context and evolves as the context evolves.

Let us define the Security Fabric by building a control system.

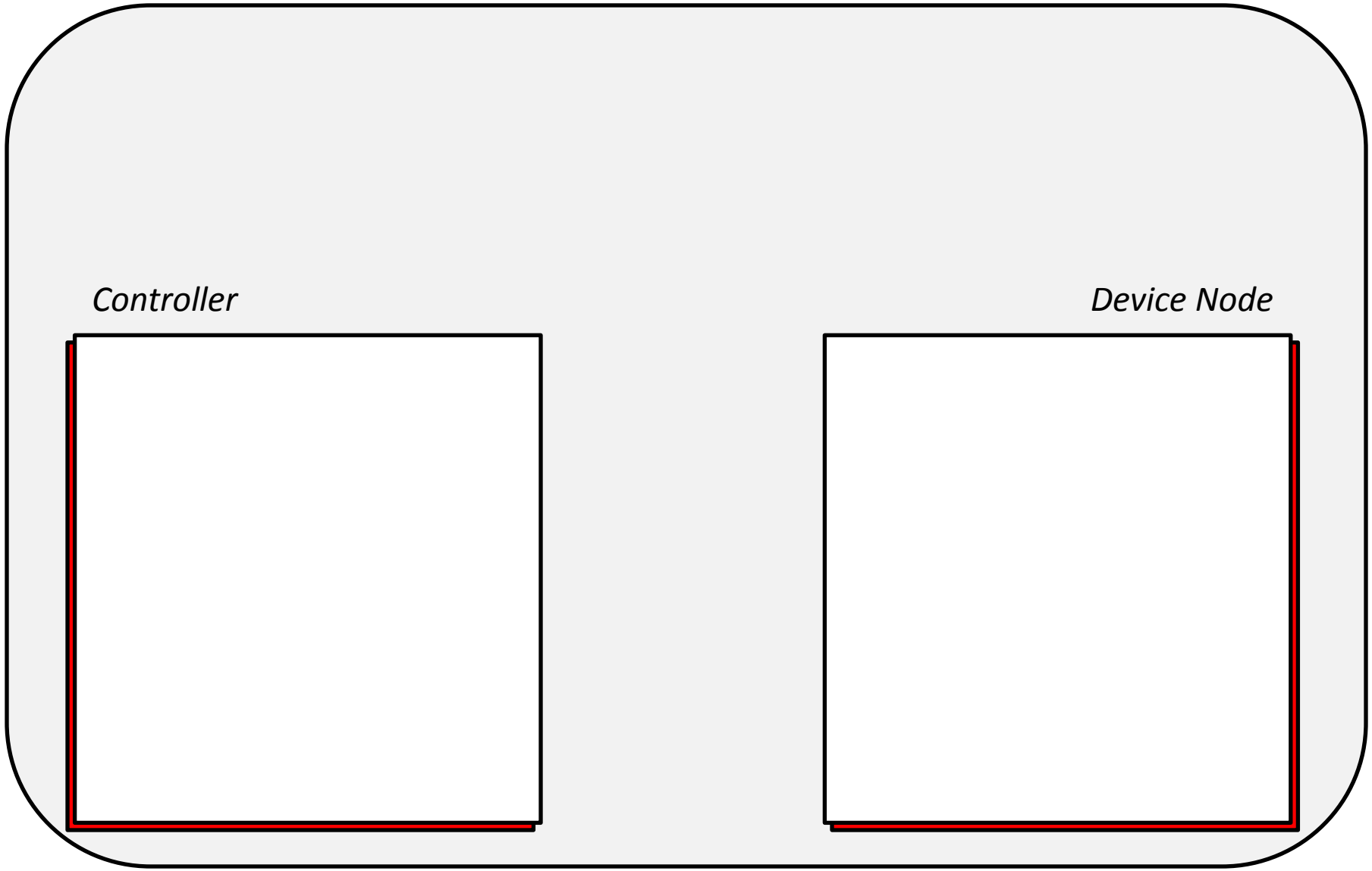
An example of a tailored trustworthy space built using the **Security Fabric** components:

In a control system, there are a controller and several devices controlled by remote device nodes.



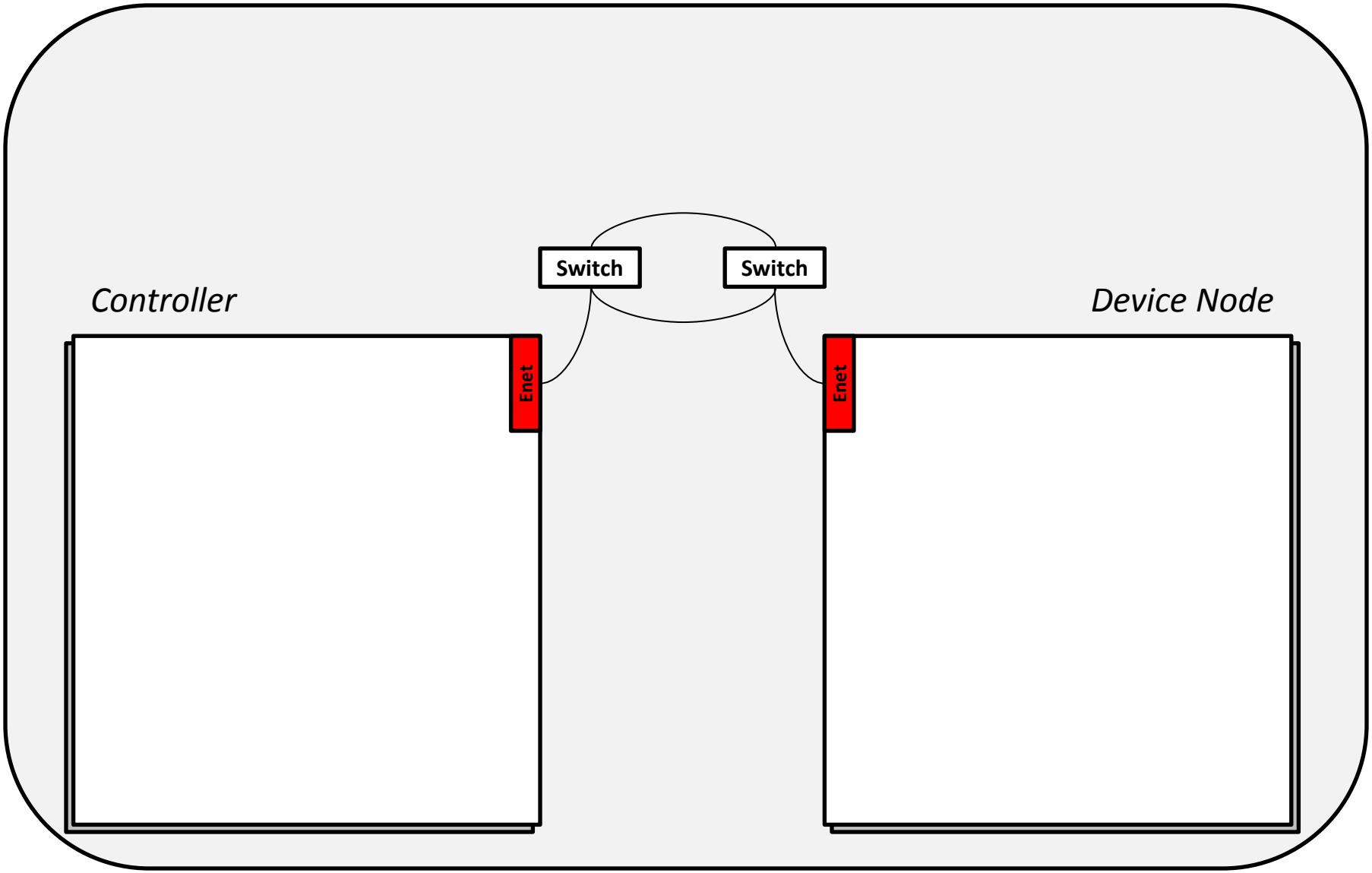
An example of a tailored trustworthy space built using the **Security Fabric** components:

Sometimes they are redundant for high availability.



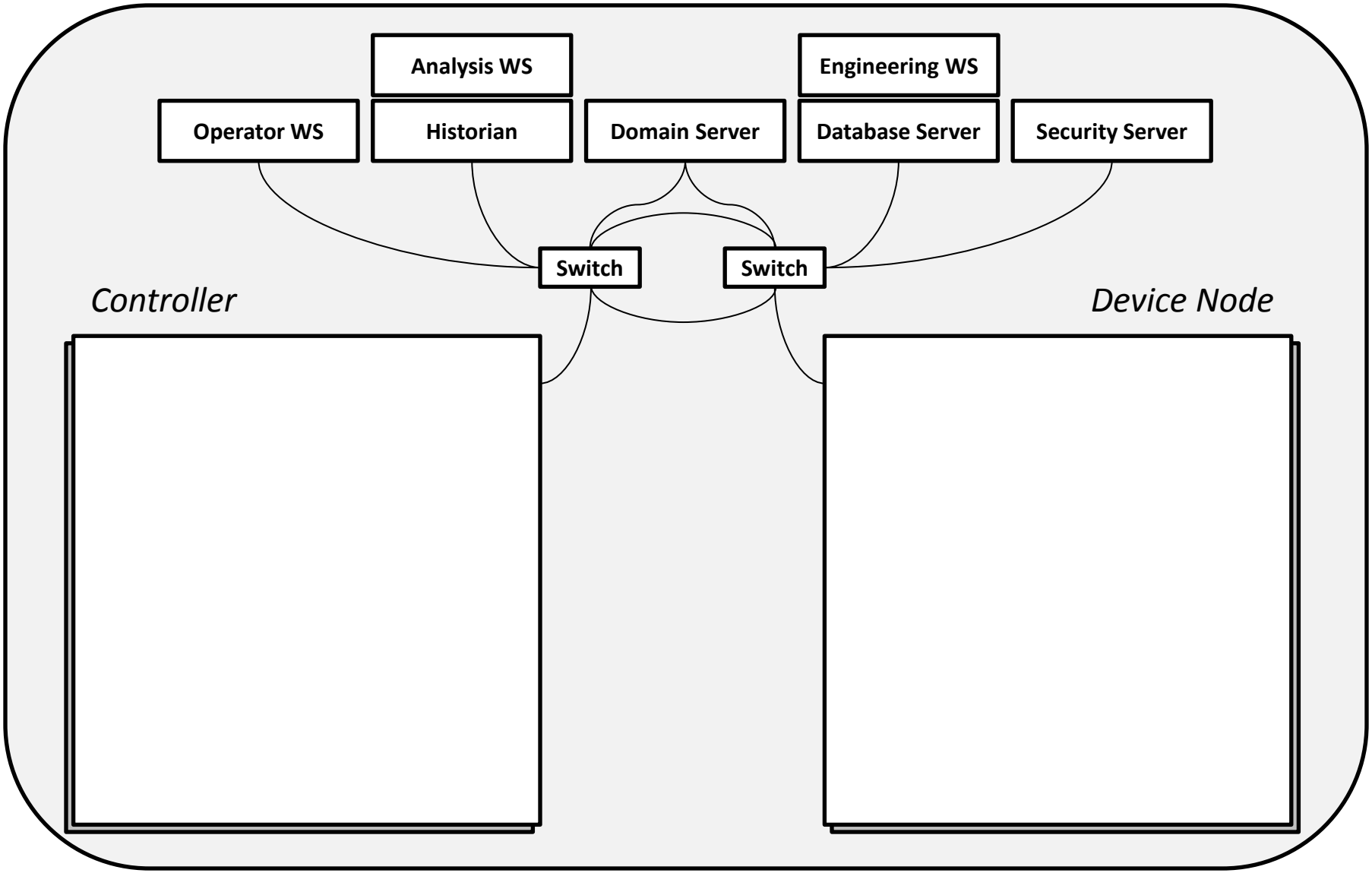
An example of a tailored trustworthy space built using the **Security Fabric** components:

They talk to each other using IP-based switches.



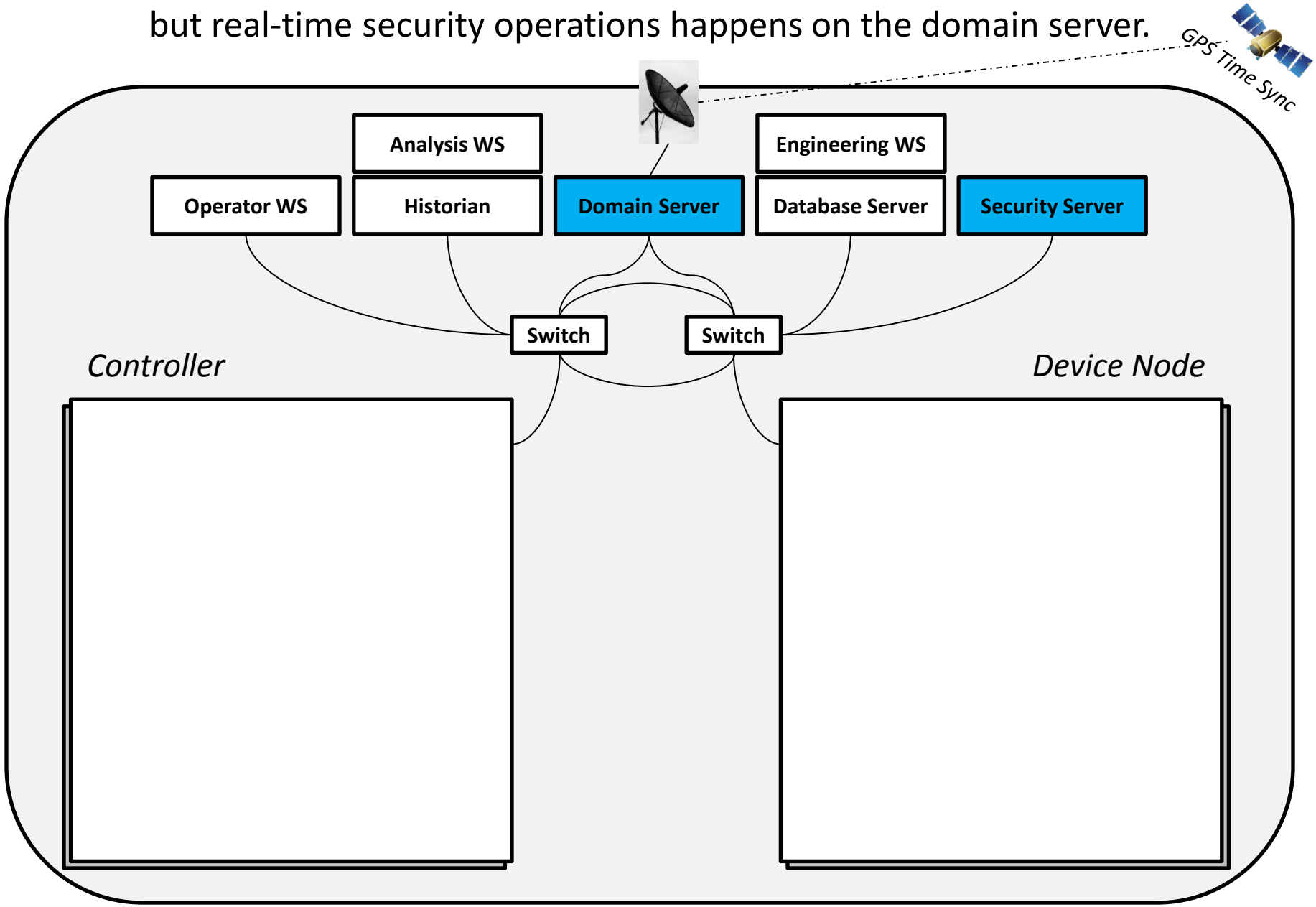
An example of a tailored trustworthy space built using the **Security Fabric** components:

They have management workstations and servers that supervise the controller and device nodes.



An example of a tailored trustworthy space built using the **Security Fabric** components:

Security management is administered on the security server –
but real-time security operations happens on the domain server.

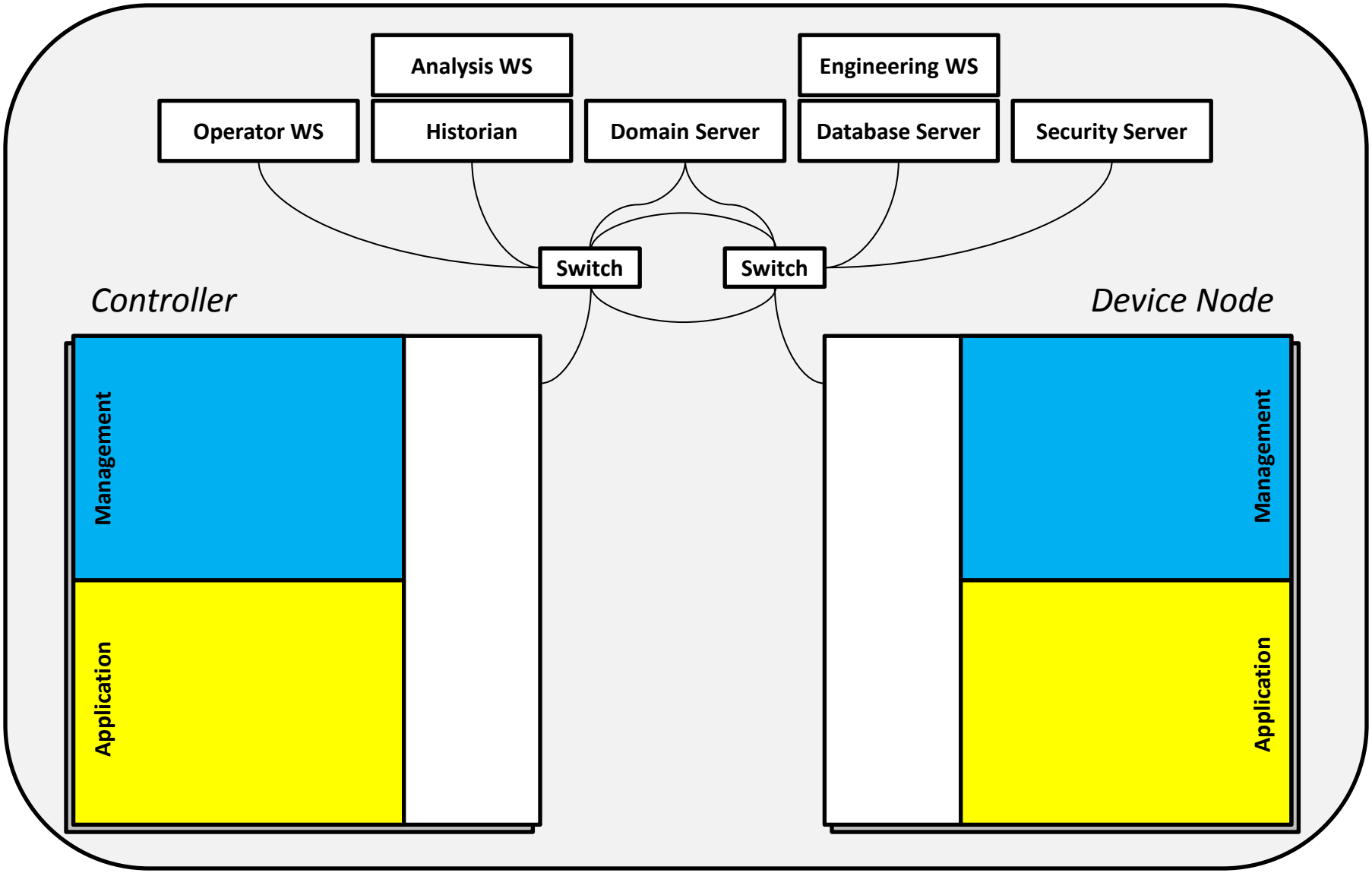


An example of a tailored trustworthy space built using the **Security Fabric** components:

The Security Fabric permeates
the distributed management functions,
but is mostly separate from the application functions.

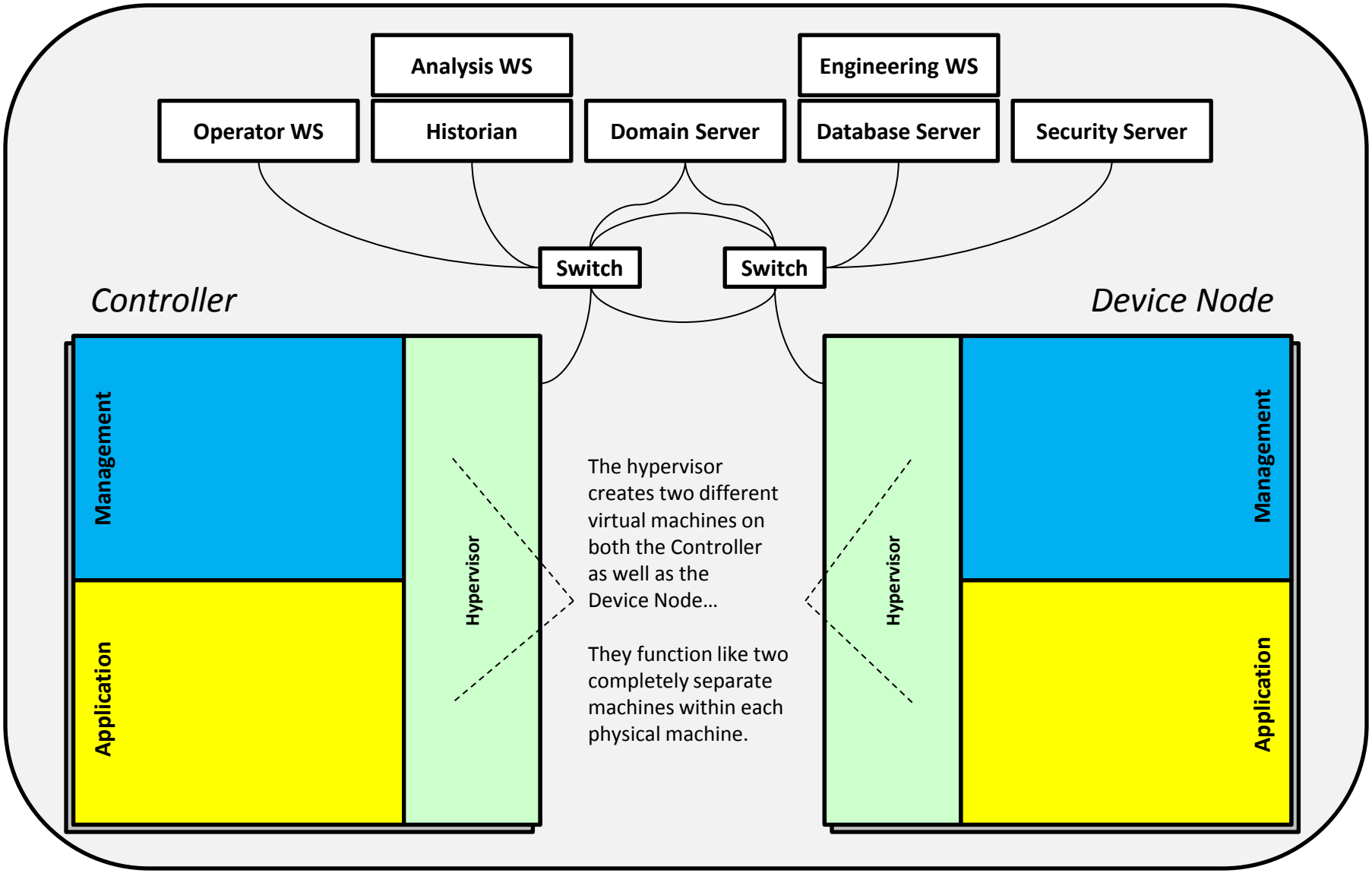
Our strategy is to separate the management functions
from the application functions as much as possible…
so that if the application becomes compromised or inoperable,
the management system can easily be used to remediate the problem.

With this in mind, both the Controller and the Device Node keep the management functions separate from the application.



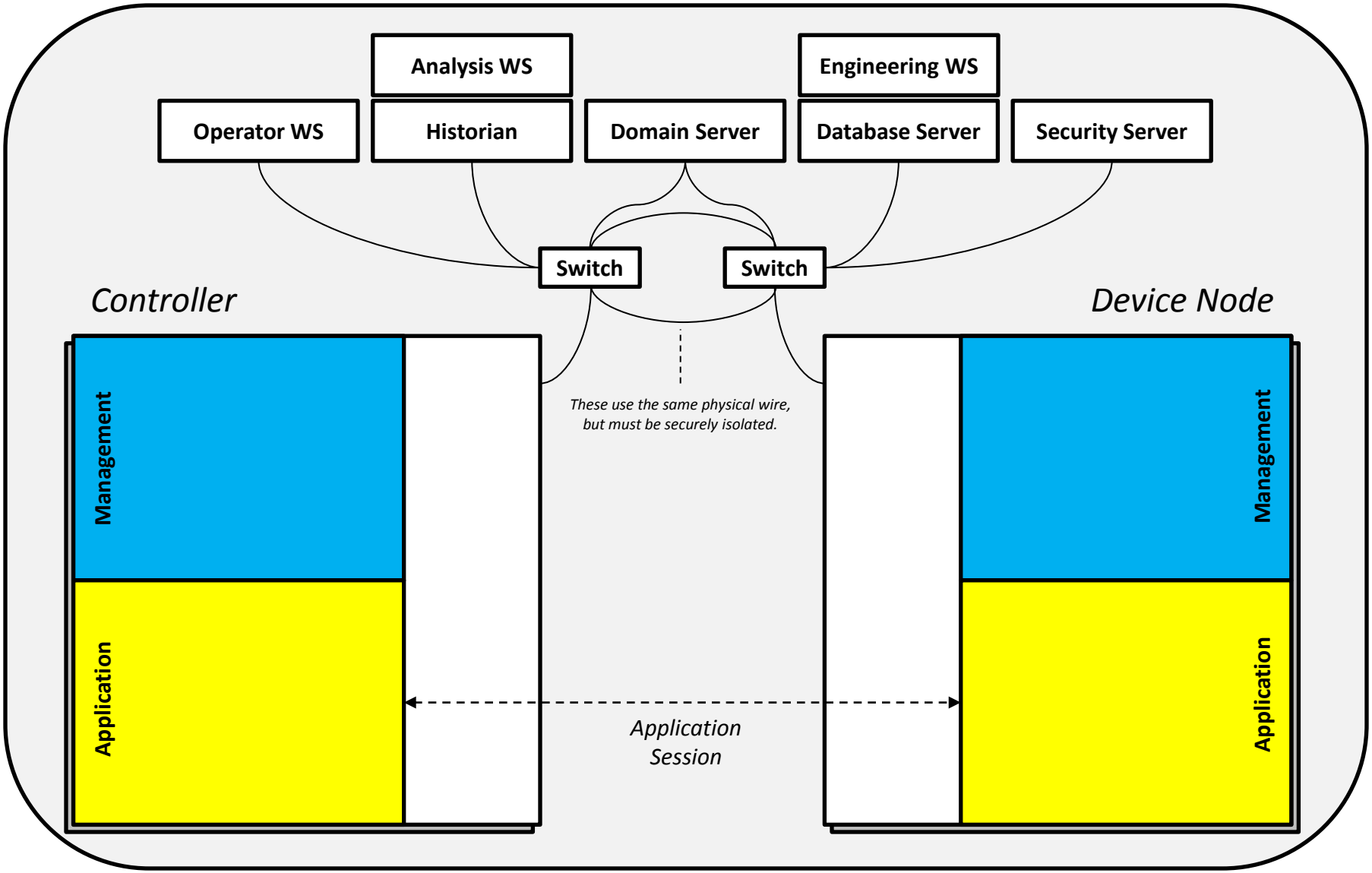
An example of a tailored trustworthy space built using the **Security Fabric** components:

This is done using a separation kernel to keep the application from ever interfering with the management functions.



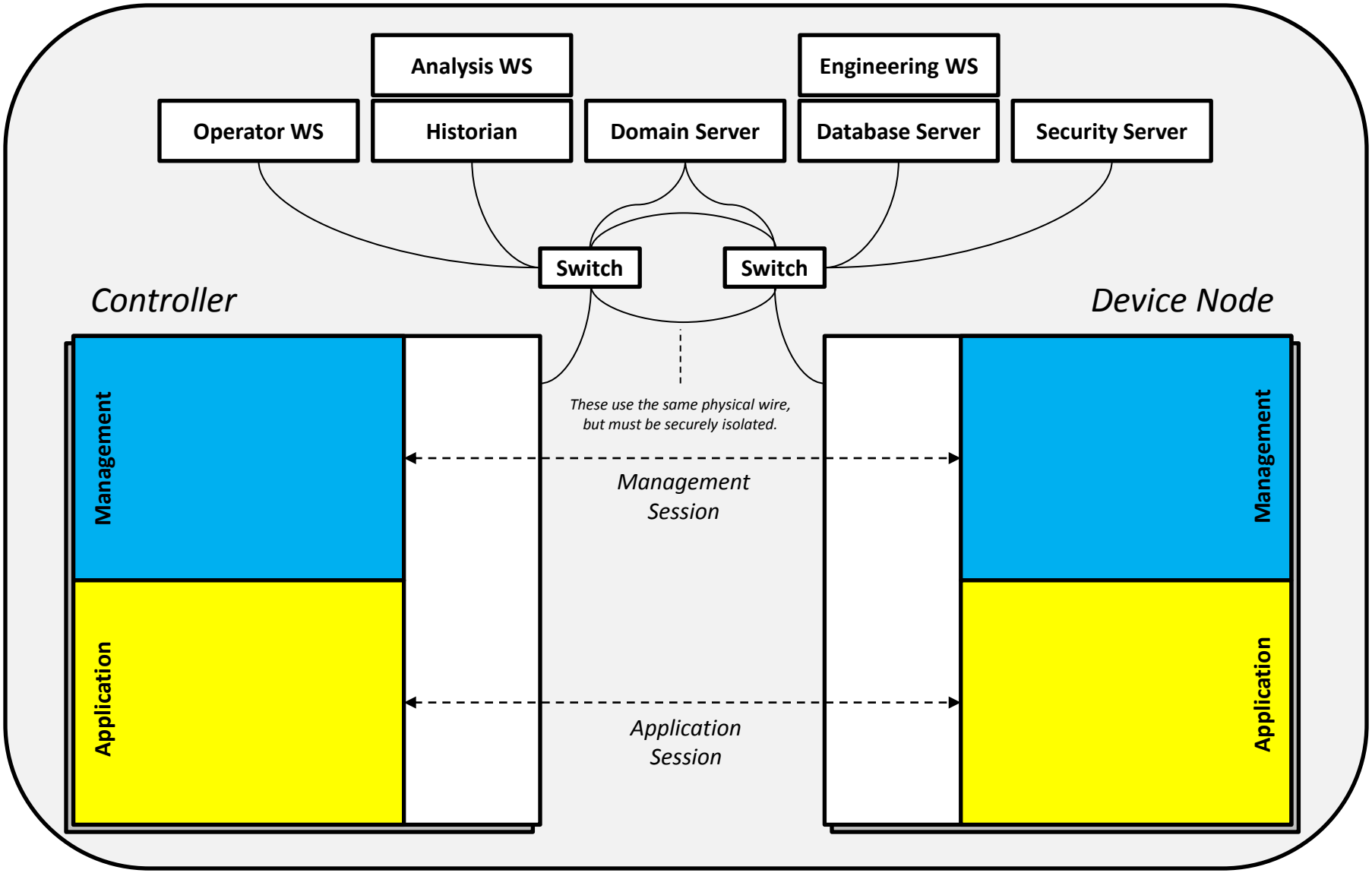
An example of a tailored trustworthy space built using the **Security Fabric** components:

The application in the controller monitors and controls the application in the device node.



An example of a tailored trustworthy space built using the **Security Fabric** components:

And the management functions and policies in the controller supports the management agent in the device node.



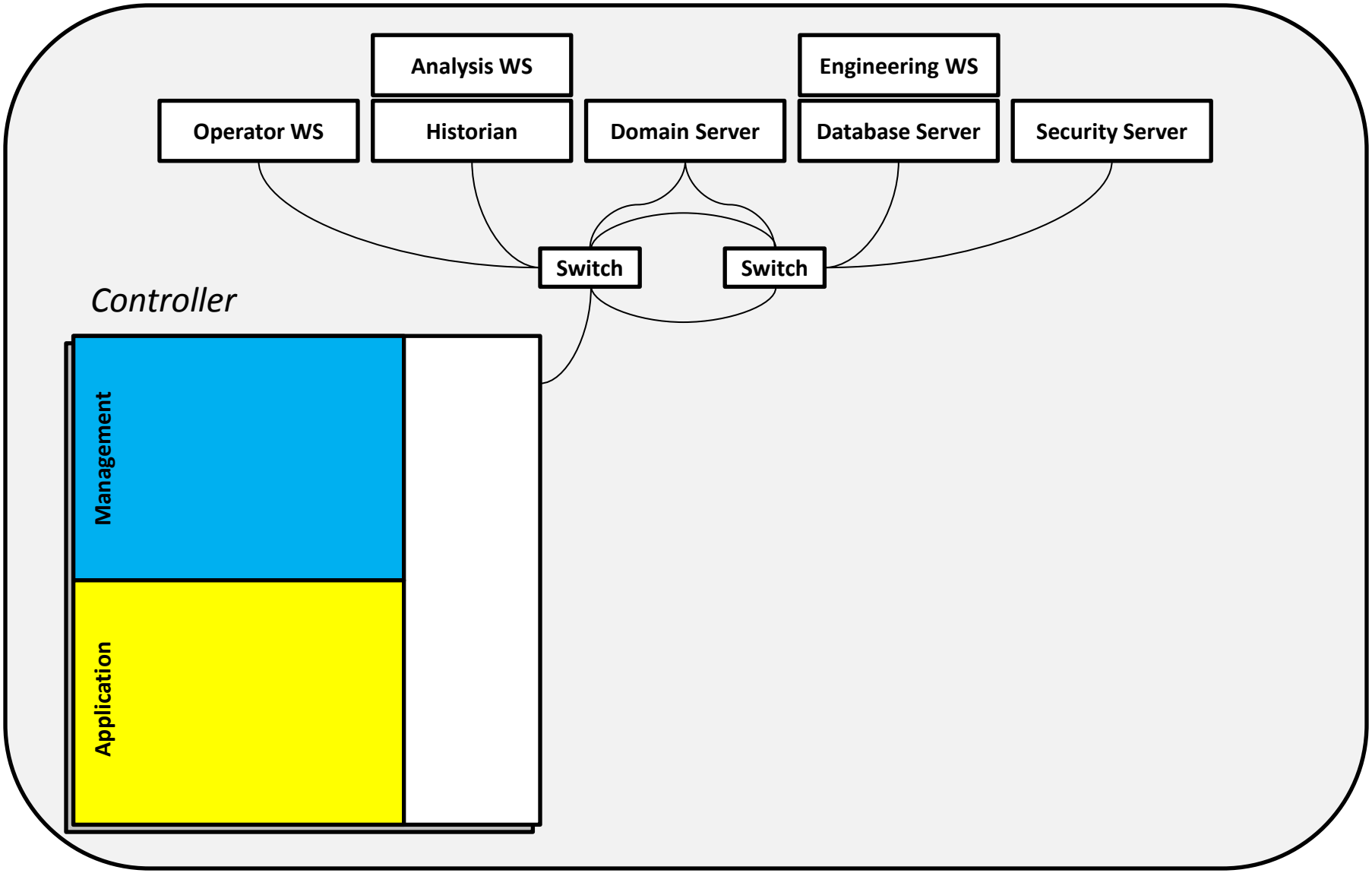
An example of a tailored trustworthy space built using the **Security Fabric** components:

These are the seven tenets of security as described in the NIST-IR 7628 Guidelines.

1. Identity Management
   - Ensures the device identity is established genuinely
2. Mutual Authentication
   - Allows both the Device Node and the Controller to verify the trustworthiness their identity to each other.
3. Authorization
   - Manages permission to proceed with specific operations.
4. Audit
   - **Records noteworthy events for later analysis**
5. Confidentiality
   - Encrypts sensitive data for matters of privacy.
6. Integrity
   - Ensures that messages have not been altered.
7. Availability
   - Prevents denial of service attacks

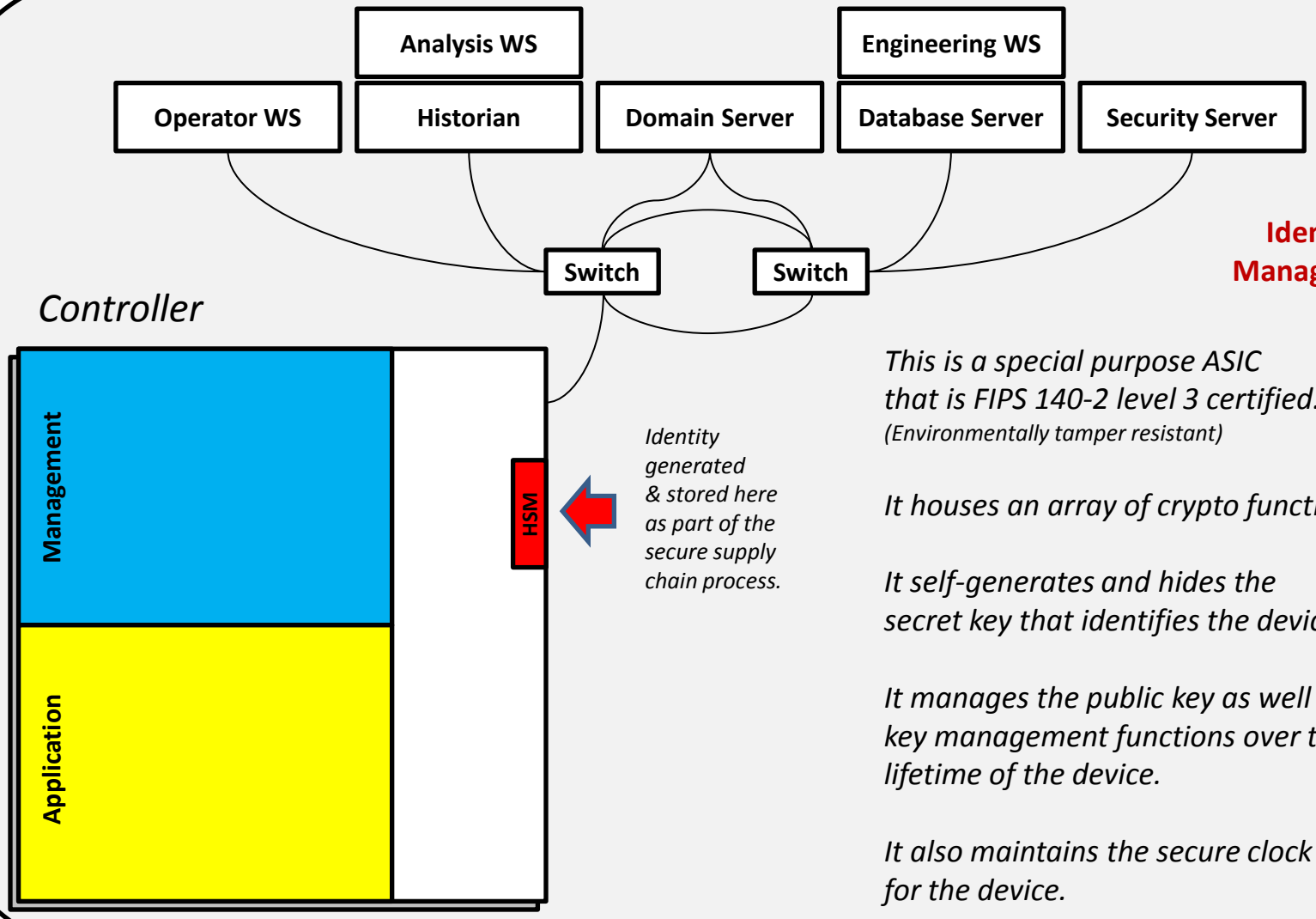
To establish the secure communications from the Controller to the Device Node using the Security Fabric elements, let us proceed in chronological order.

The Controller must power on before any of the device nodes can use it.



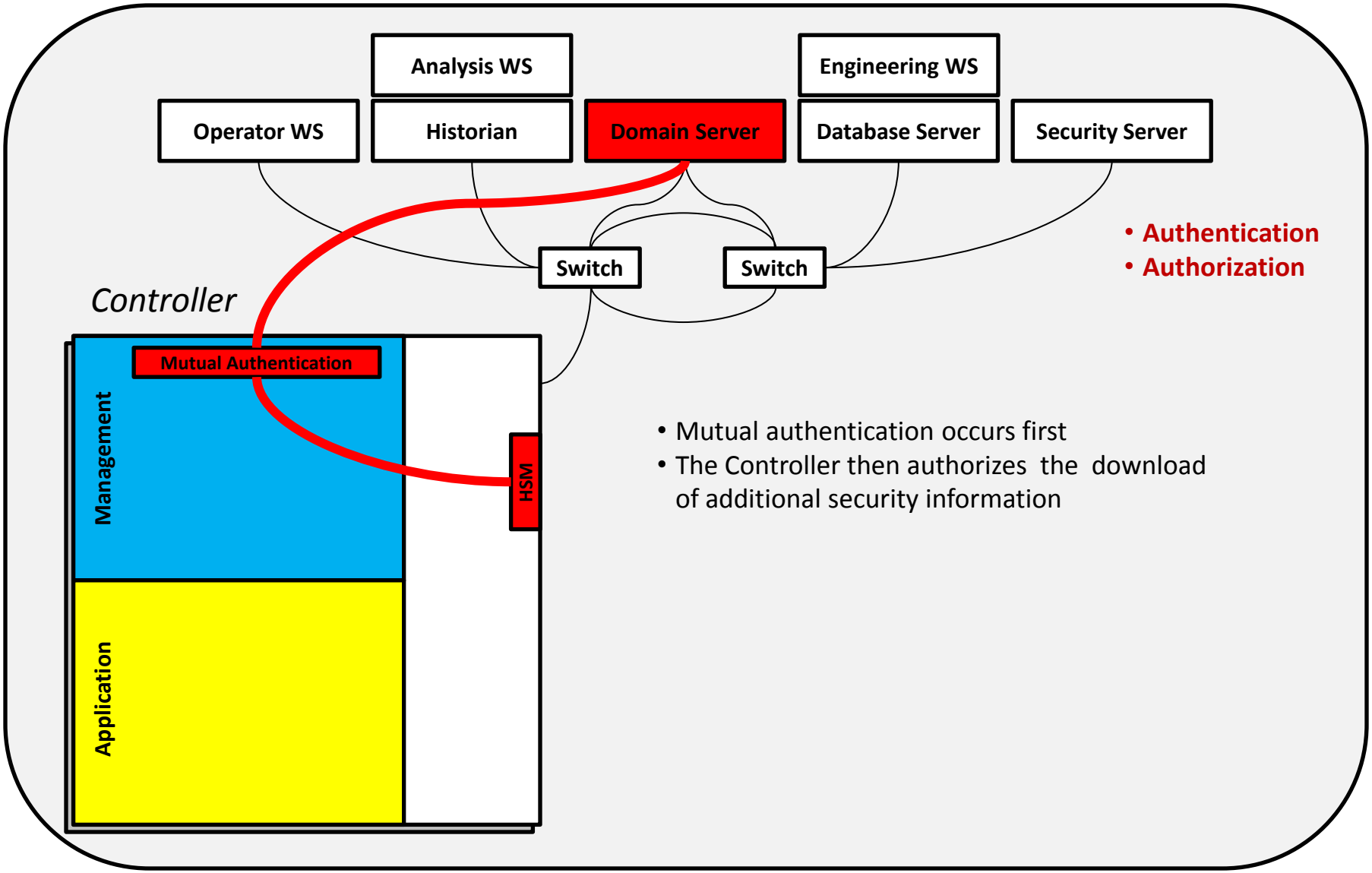
An example of a tailored trustworthy space built using the **Security Fabric** components:

Identity Management is the most crucial aspect of embedded security – we use a Hardware Security Module to protect the unique identity of the Controller.



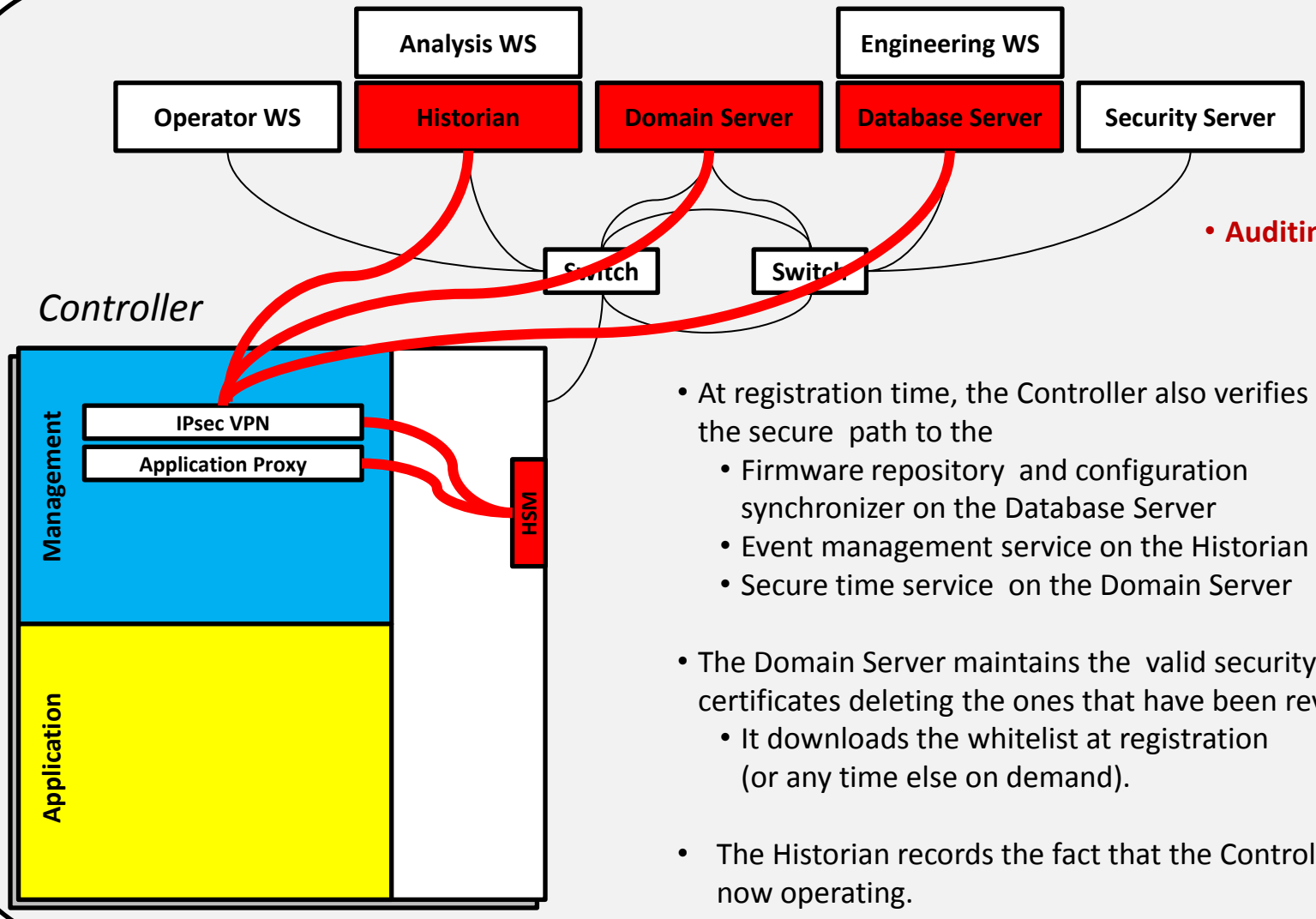
An example of a tailored trustworthy space built using the **Security Fabric** components:

Step two is to use the secure identity to mutually authenticate and get credentials from the Domain Server that uses Active Directory and its Kerberos PKINIT service meant to support embedded devices.



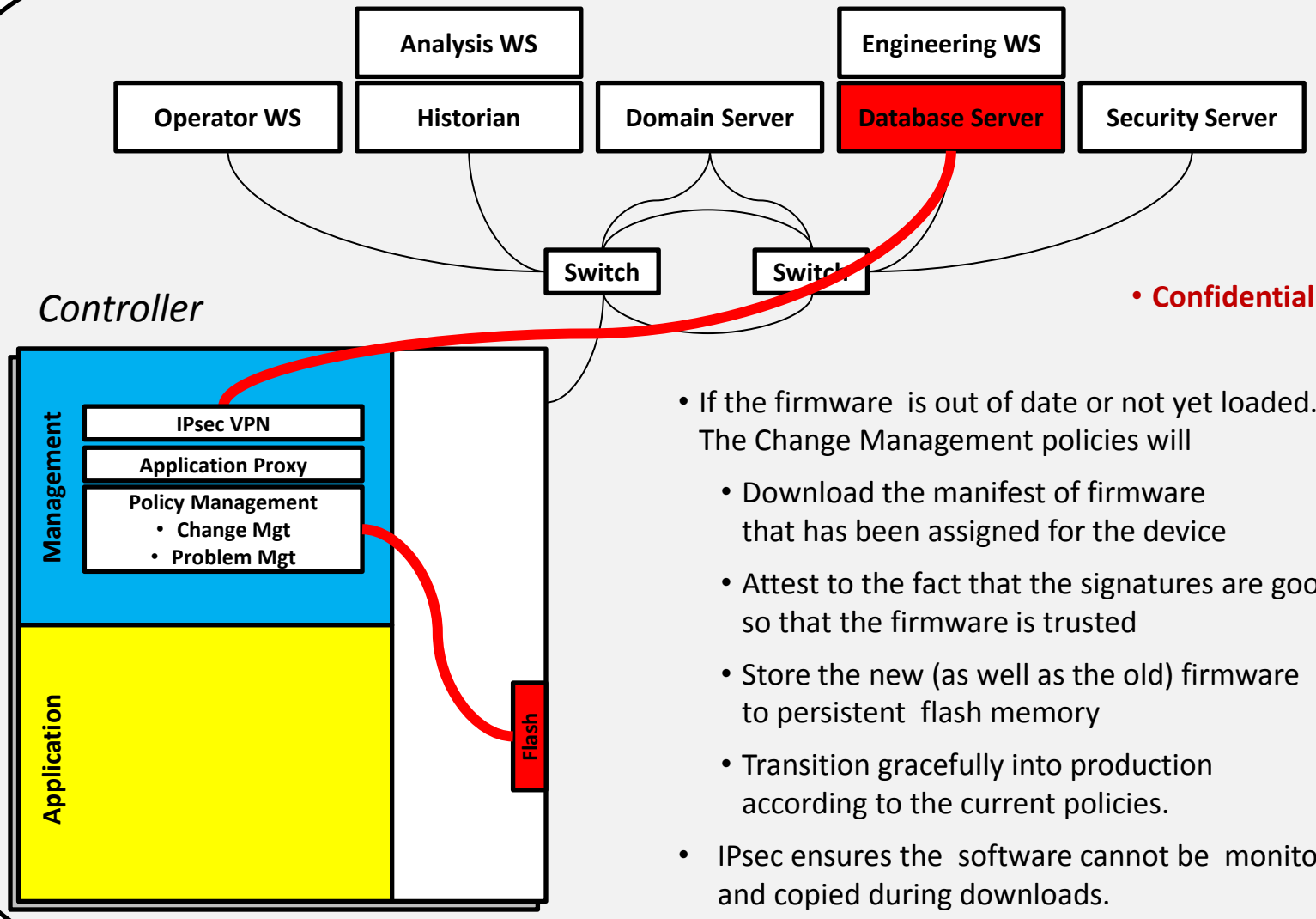
An example of a tailored trustworthy space built using the **Security Fabric** components:

Step three is to use the secure credentials exchange to determine the authentic paths to important management servers, and to download the up-to-date whitelist.



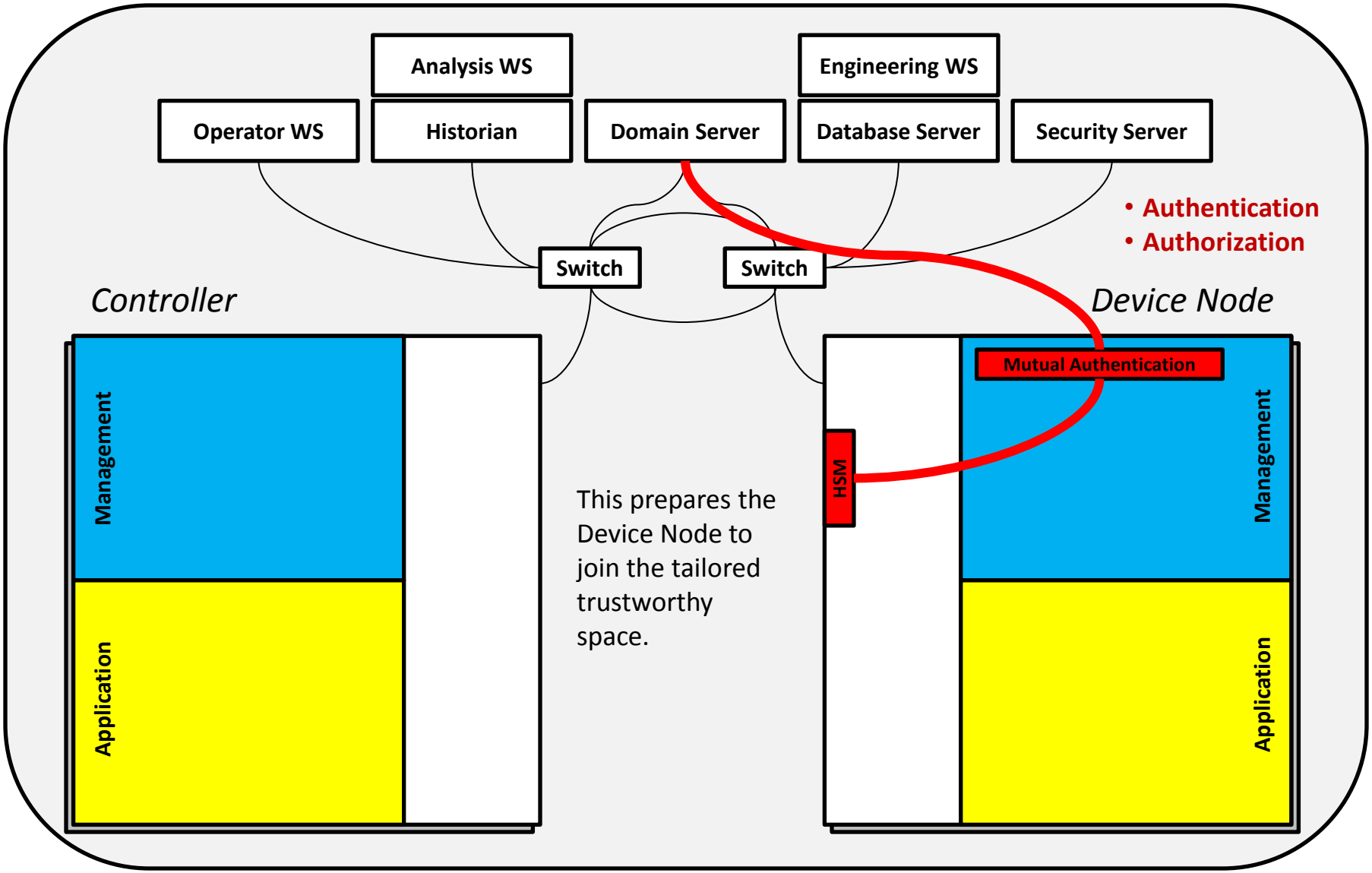
- At registration time, the Controller also verifies the secure path to the
  - Firmware repository and configuration synchronizer on the Database Server
  - Event management service on the Historian
  - Secure time service on the Domain Server
- The Domain Server maintains the valid security certificates deleting the ones that have been revoked
  - It downloads the whitelist at registration (or any time else on demand).
- The Historian records the fact that the Controller is now operating.

Step four is to update the firmware to the latest rev if it is out of date.



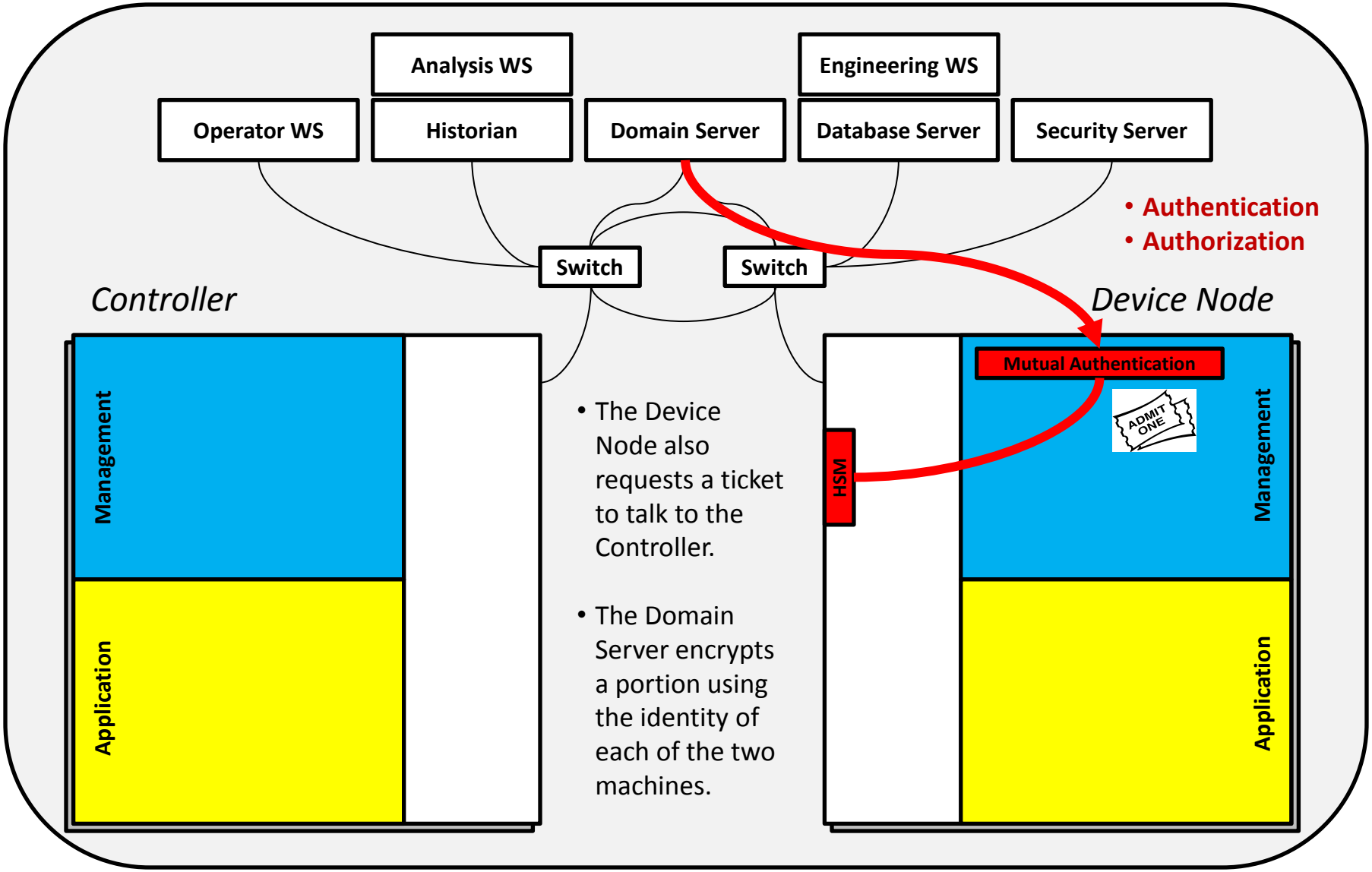
An example of a tailored trustworthy space built using the **Security Fabric** components:

All Device Nodes that want to be part of the Security Fabric must also authenticate with the Domain Server (the trusted third party) whenever they power up.



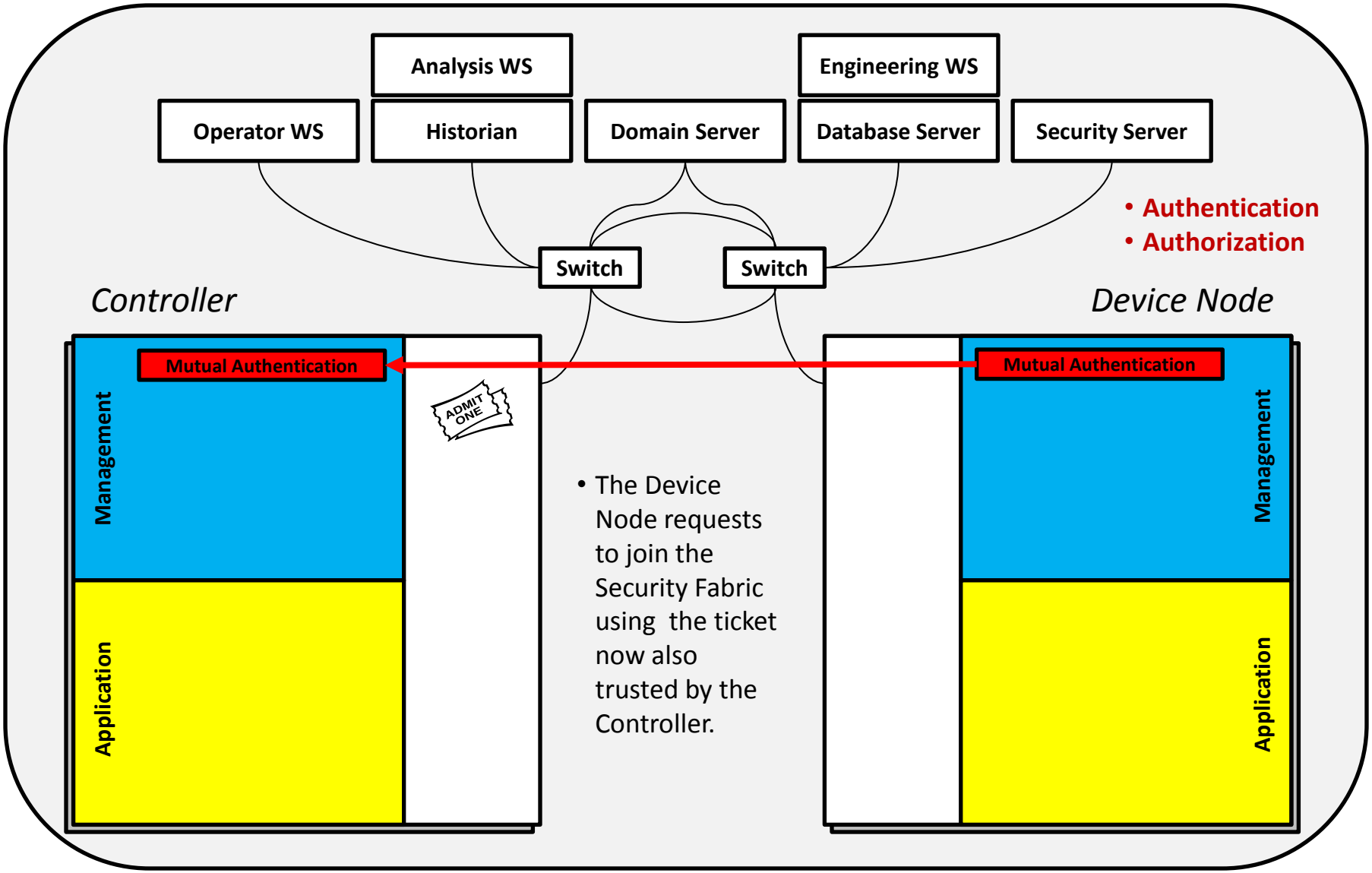
An example of a tailored trustworthy space built using the **Security Fabric** components:

The authentication ticket received from the Domain Server contains a section encrypted by the Device Node public identity key plus a section encrypted by the Controller public identity key.



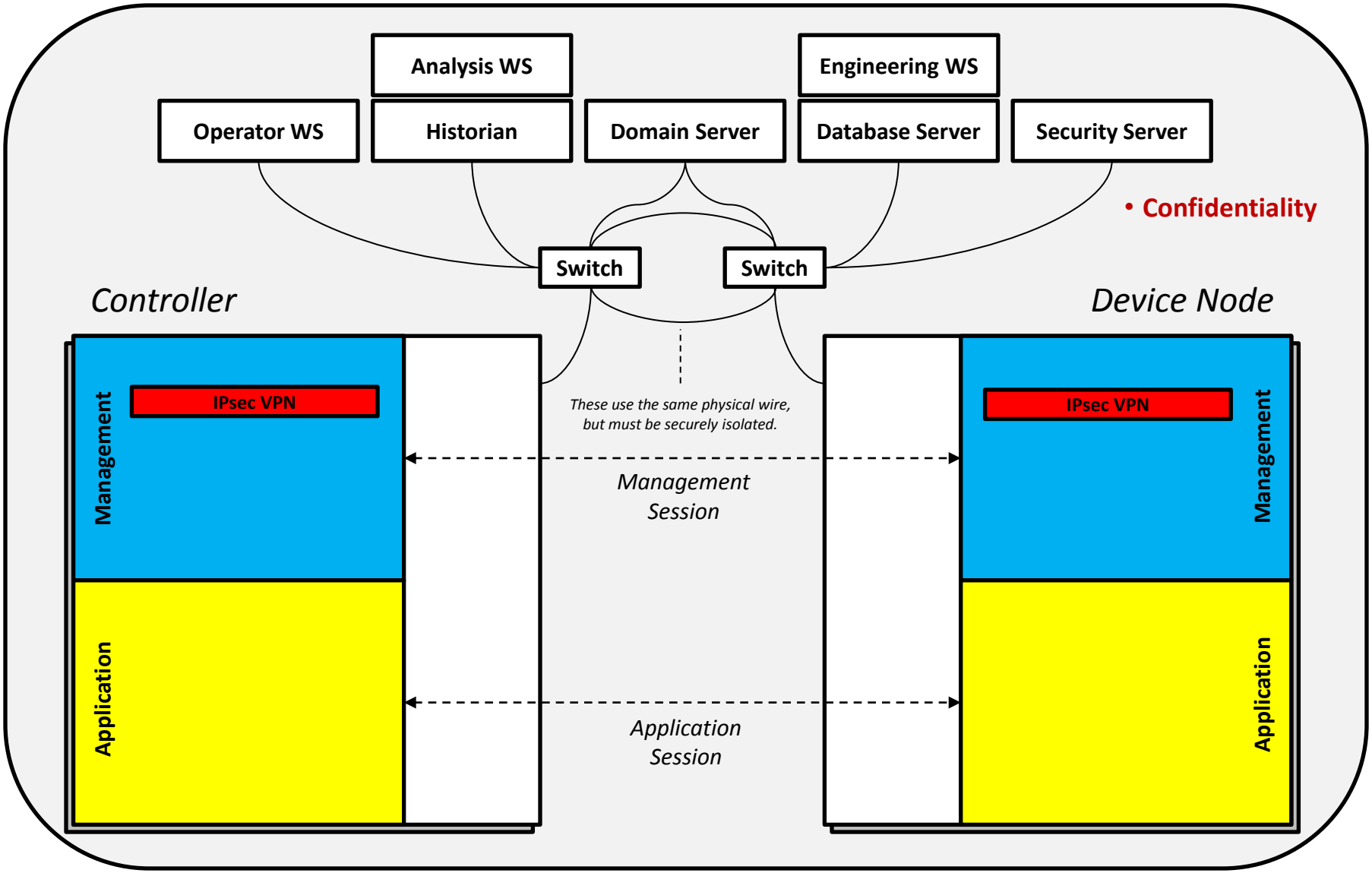
An example of a tailored trustworthy space built using the **Security Fabric** components:

The next step is for the Device Node to establish secure communications with the Controller.



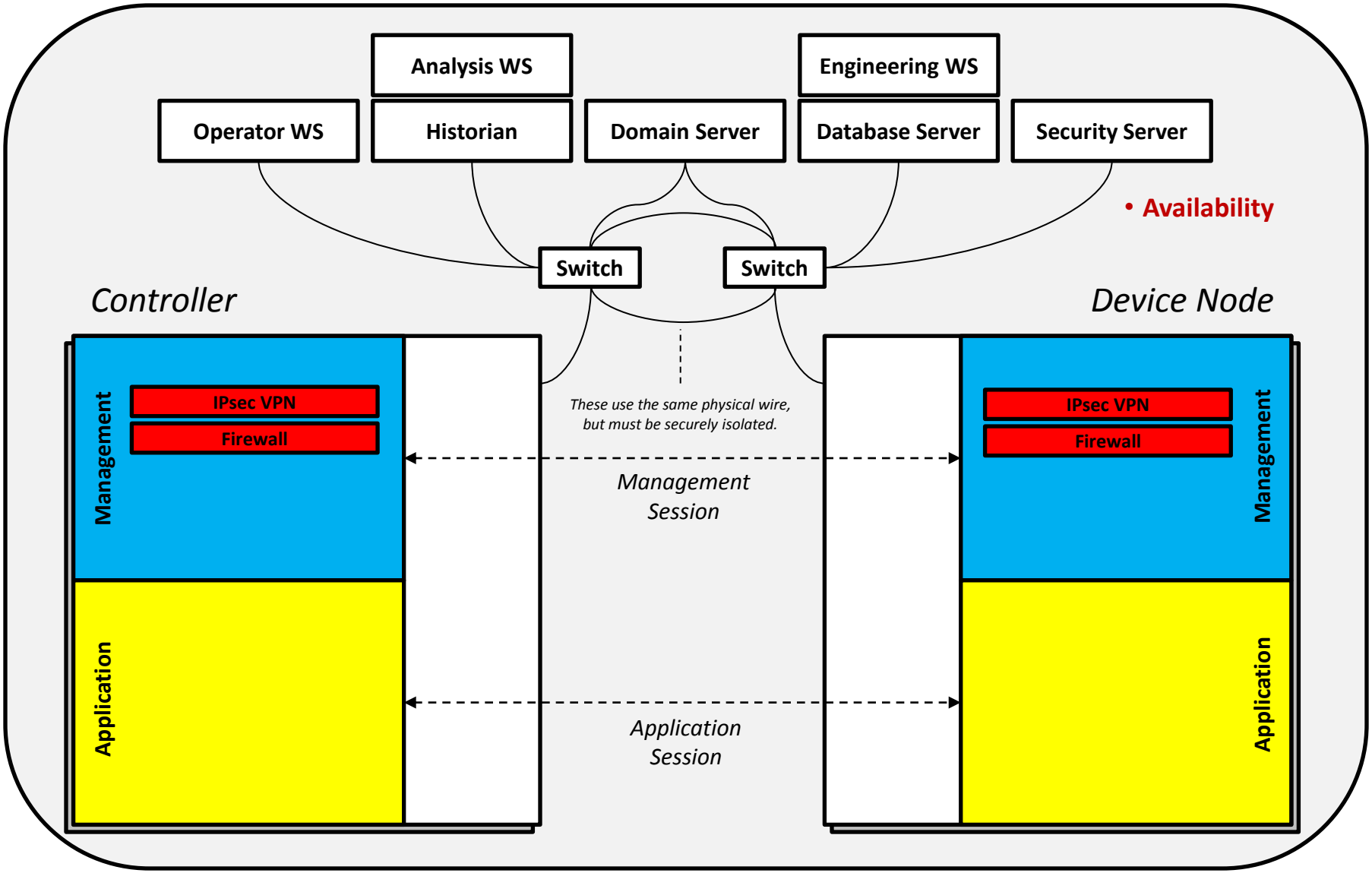
An example of a tailored trustworthy space built using the **Security Fabric** components:

Once authenticated, the device node can proceed to establish two secure paths to the Controller: one for management purposes and one for application purposes.



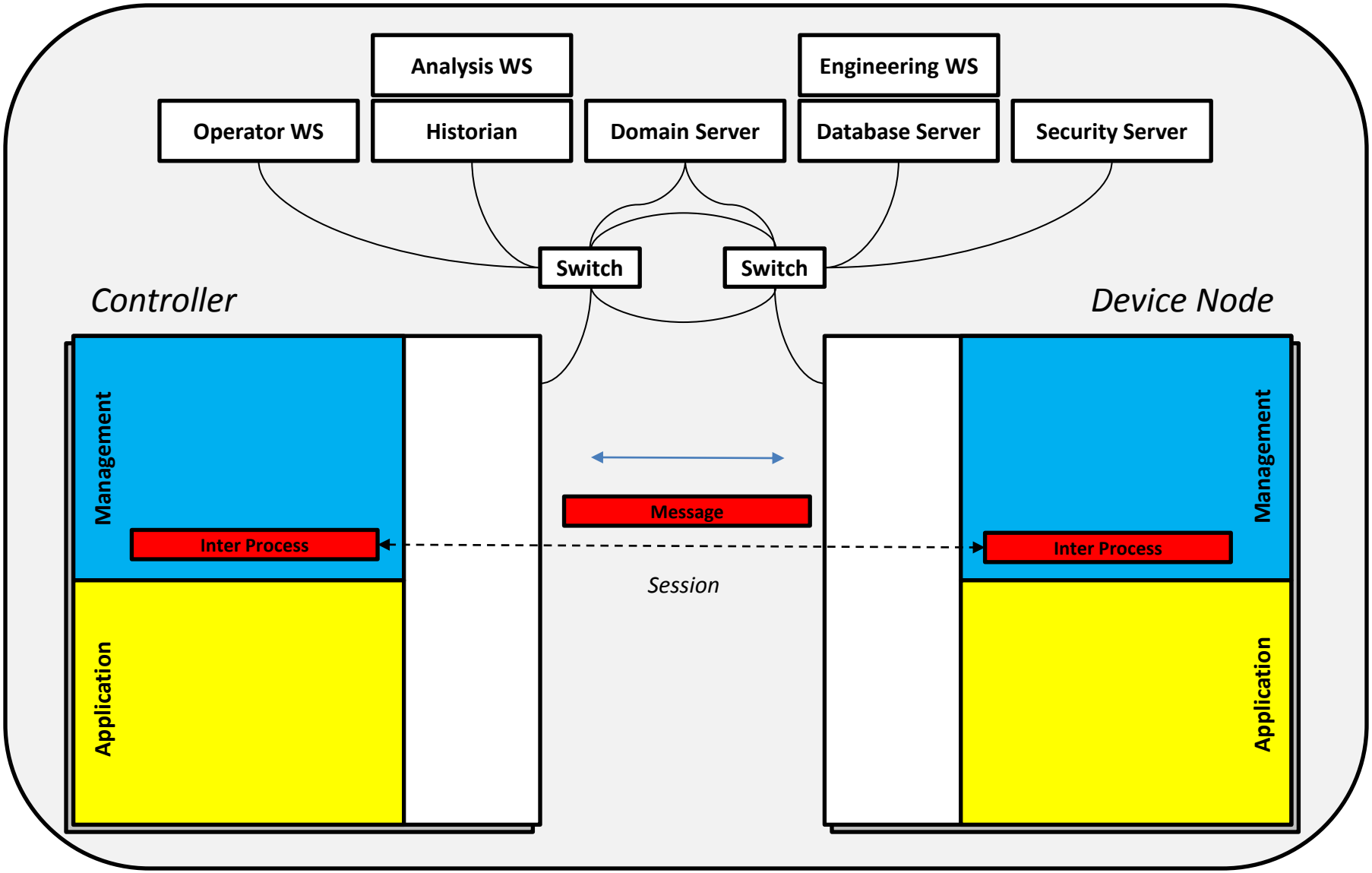
An example of a tailored trustworthy space built using the **Security Fabric** components:

The small embedded firewall in the communications path
protects against denial of service attacks
as well as a number of sophisticated malware attacks.



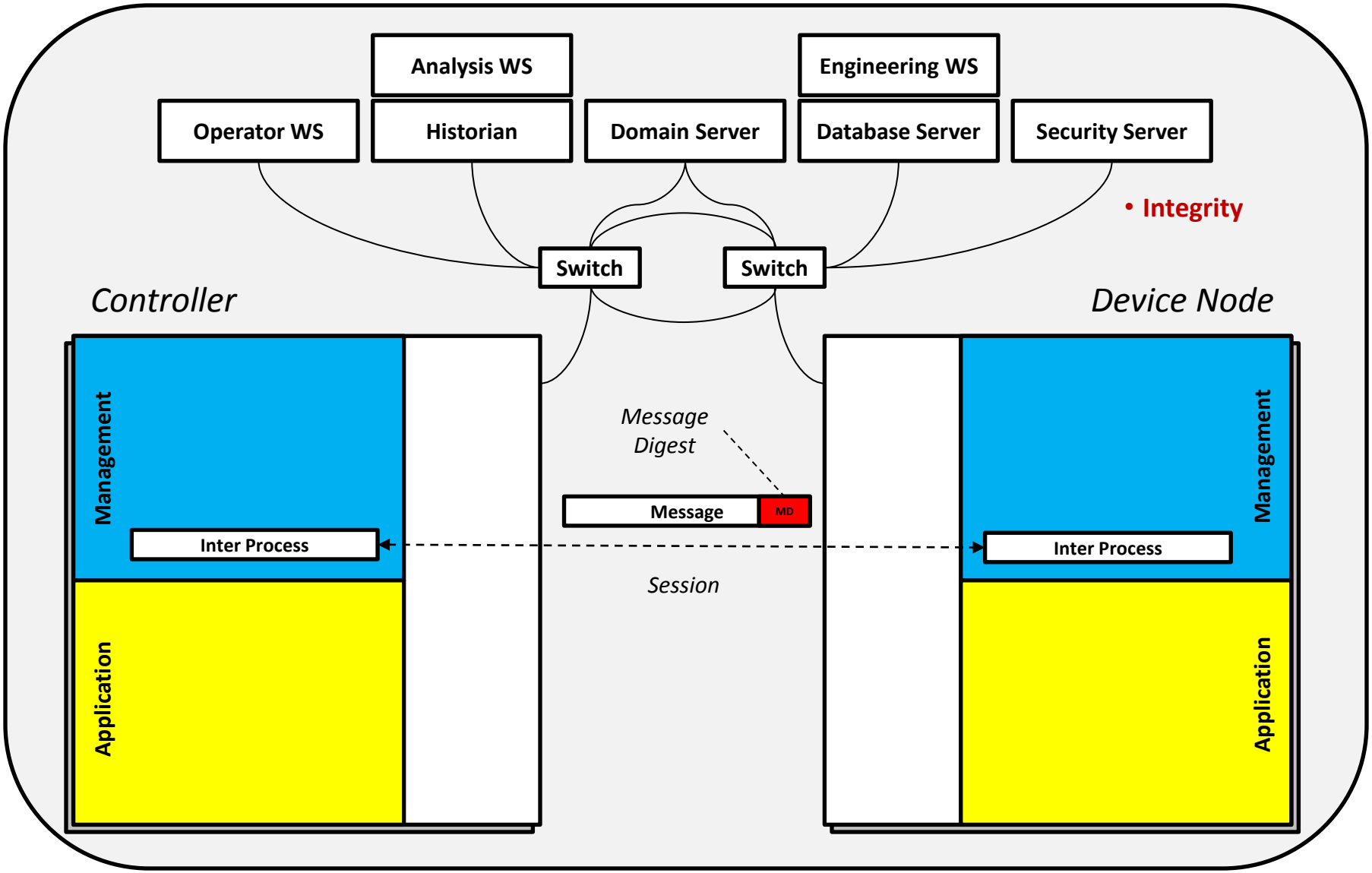
An example of a tailored trustworthy space built using the **Security Fabric** components:

The inter-process communications services of the middleware uses messages to communicate back and forth between the Controller and the Device Node over the secure sessions.



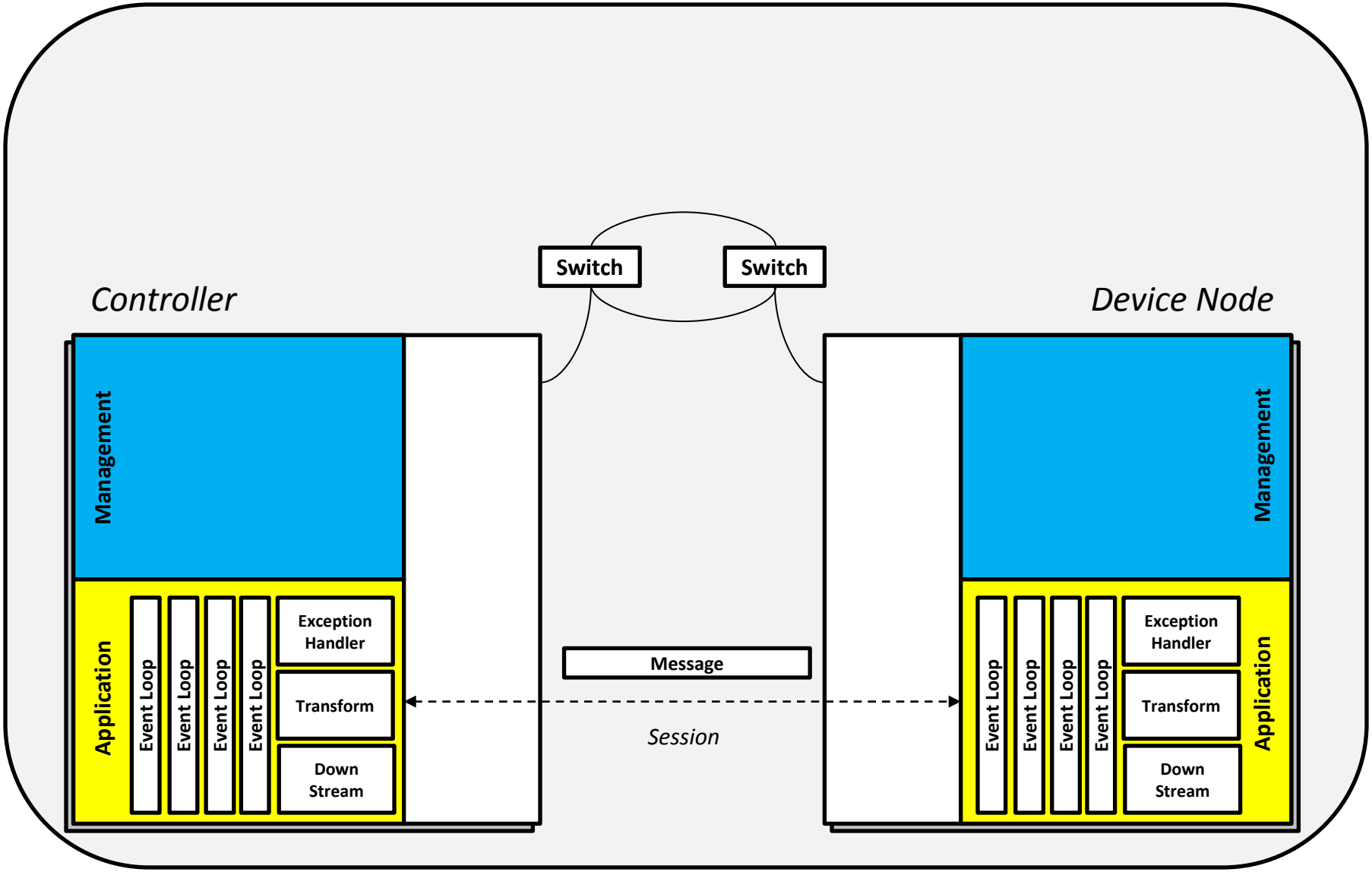
An example of a tailored trustworthy space built using the **Security Fabric** components:

The inter-process communications services computes a secure message digest and appends it to the end of each message to ensure that the message is never altered in flight.



An example of a tailored trustworthy space built using the **Security Fabric** components:

So now, the Controller and the Device Node can commence doing real work without ever having to think about the security aspects of the system.



An example of a tailored trustworthy space built using the **Security Fabric** components:

In Summary,

Security Fabric

provides the features for embedded security based on the NIST-IR 7628 guidelines.

It also provides a framework for a tailored trustworthy space.

Kerberos is an integral element
for the private network side of smart grid security.