IIoT- Predictive Maintenance using a Thermal Imaging Camera Using LF Edge’s Fledge and EVE

THE LINUX FOUNDATION

Use Case

- Many times the health of a device on a factory floor can be determined by its operating temperature
- Manual Monitoring-
  - Monitoring can be done manually by an operator, but that has many limitations: 1. operator must be physically be there 2. no continuous coverage 3. no predictive component
- Continuous/ Automatic Monitoring-
  - Monitoring can be done by connecting the thermal imaging camera to a local computer/server but there are many issues with this: 1. connectivity might be poor 2. Older systems might be present, and the owner needs to keep them because of previous investment.

UC- Needs

- A system that allows:
  - Continuous monitoring of the system
  - Able to react to an out of bounds condition (i.e. too hot)
  - Remote monitoring of the system
  - Able to send data to another system (OSI's Pi server, historian, MS Azure Cloud, AWS, Google Cloud)
  - No touch maintenance of the system (remote updating, monitoring)
  - Security of the data and the components
  - ML both near the edge (or on) and in the cloud

UC- Solution EVE and Fledge

- EVE sits on the bare metal and ensures:
  - The security of the device
  - Software updates
  - Hardware independence
- Fledge
  - Can be packaged with a VM and distributed/updated via EVE
  - API's that allow the abstraction of hardware device
  - Local processing to react to conditions

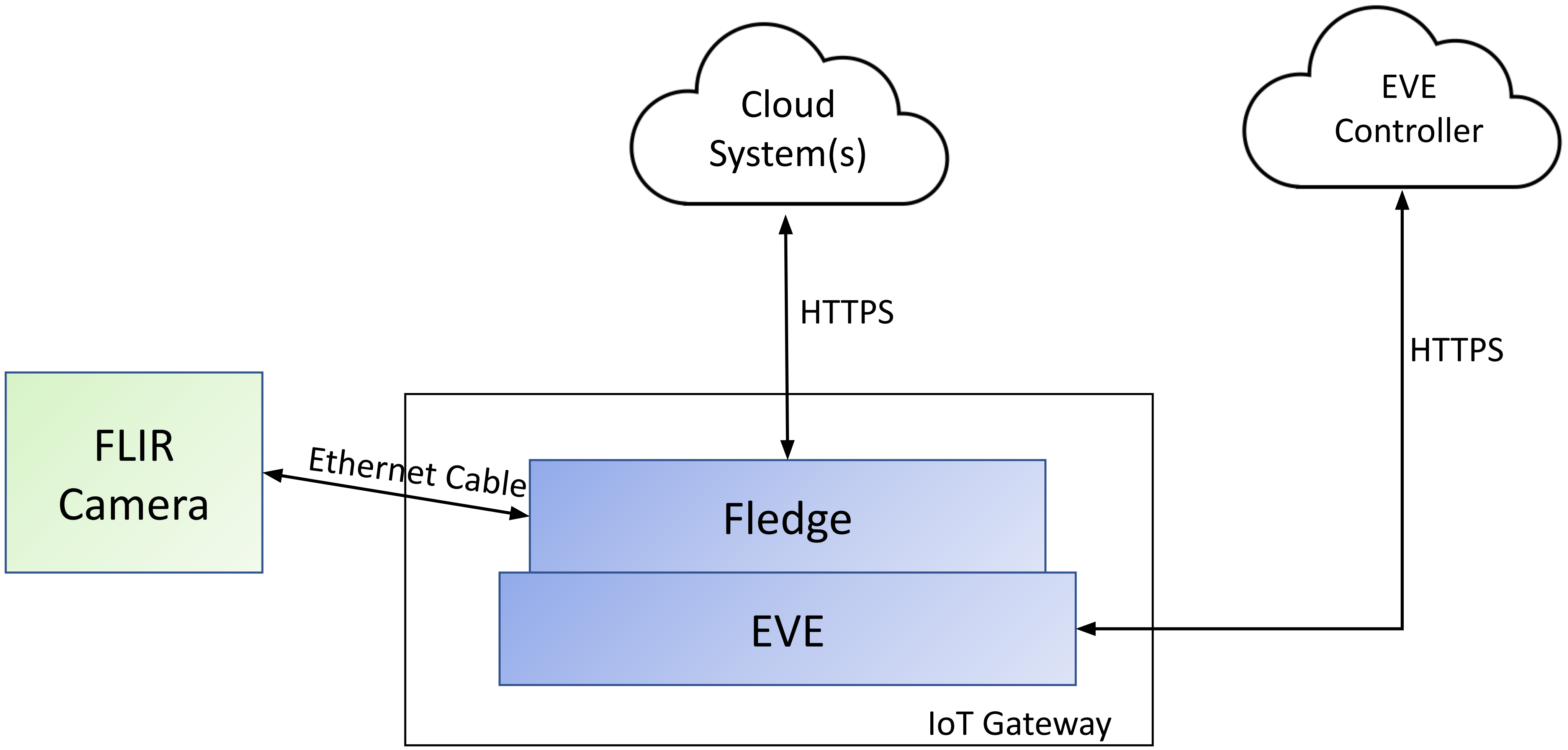
Use Case Details

Attributes	Description	Informational
Type	New	New
Industry Sector	Industrial IoT	
Business driver	Predictive Maintenance	
Business use cases	Many devices give off hints that they will need to have maintenance earlier than their schedule maintenance. Through Machine Learning (ML), we can create models that will allow us to know that a device will soon need maintenance. For many machines, we can gain a great deal of information on the health of the device by looking at the temperature of the device. This requires collecting the data and then sending it to a Historian or similar device. These data points can be sent to the cloud to be modeled. Other requirements	Predictive maintenance: There are many different types of models. For example, many models do not need to be done in real time. Thus, the data can be sent to the Cloud and processed. The data is not time critical, so if there is a delay in sending/receiving data, the data will need to be stored and then sent when the network is available.

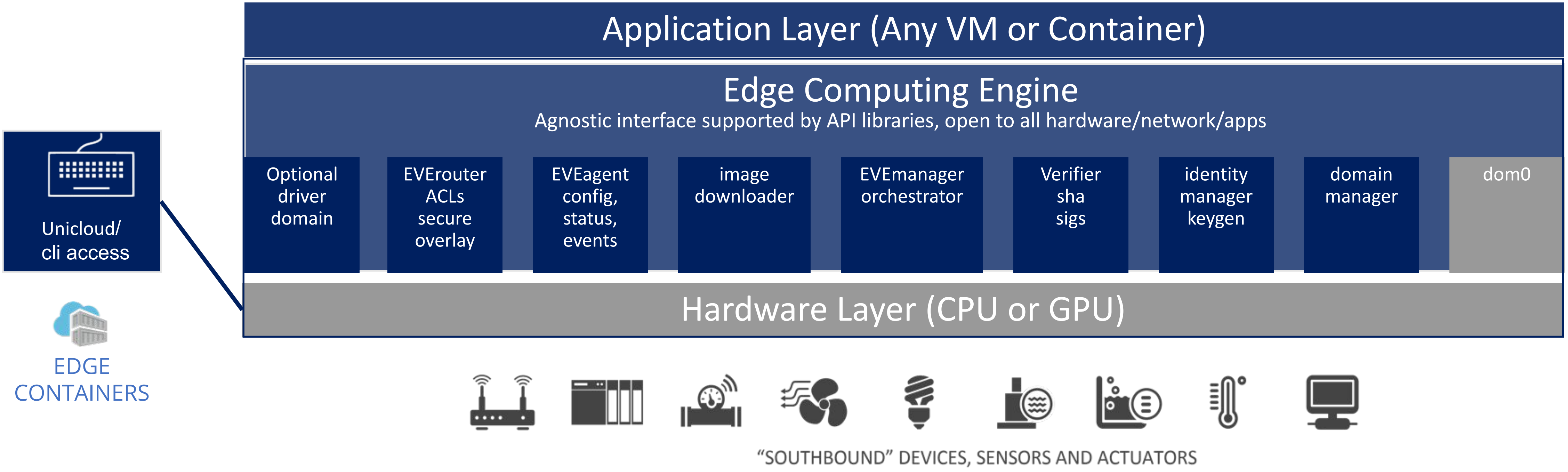
Use Case Details

Attributes	Description	Informational
Business use cases (Continued)	Other requirements •Need to take the current temperature of the device and react in near real time to rising temperature • Example: If over 150 C- send out a warning to a email list, show warning on a UI if over 180 C trigger light or horn if over 200 C trigger shutdown process Other variations: Monitoring restricted spaces •If a human enters in a space, • first level of restriction- sound an alarm and turn on lights • second level- start shutdown process	Yet, there are many scenarios, where real time or near real time is required. An example of this would be a machine reaching a maximum temperature. As it approaches this, we would want to send out a warning and then if it reached this critical temperature, the device needs to be shut down. For this type of scenario, there needs to be a server or space on the IoT gateway that can process the data in real time.
Business Cost - Initial Build Cost Target Objective	Cost is only for the hardware-	
Business Cost – Target Operational Objective	varies widely depending on accessories. The IoT Gateway can be under $500 to over $5,000	
Security need	Because of the remoteness of the devices, need the ability to control ports (turn on/off)	
Regulations	Varies depending on local regulations	
Other restrictions		
Additional details		

Simplified Drawing of System



EVE Edge Computing Engine Architecture



Project Scope

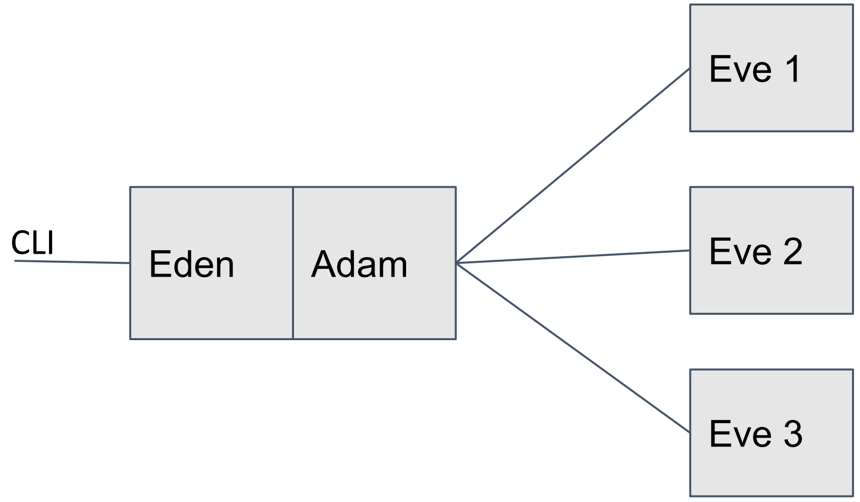
- Establish standardized Edge Container Object (ECO) format
- Build EVE edge computing engine and controller (EVC) interface
- API + CLI reference implementation

EVE SCALABILITY

Eve connects to an Adam controller bundled into Eden as a control plane

Adam + Eden allow:

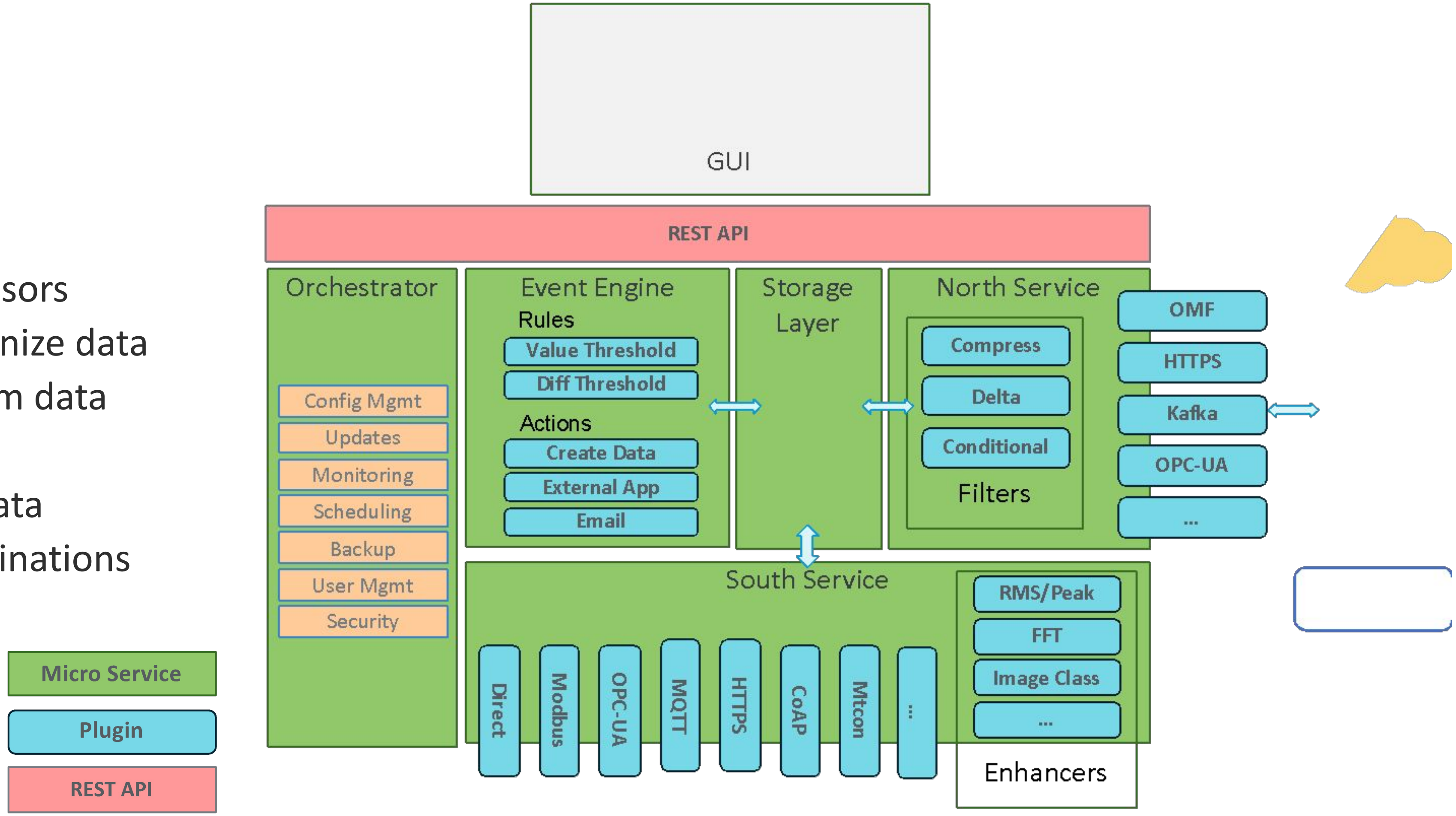
- Mass Fledge installation.
- Single control point
- Easy update and devOps
- Single logs repo



./eden --config eve1 test ./tests/flir - to deploy and run the test

ARCHITECTURE

- Collect Data - from any/all sensors
- Aggregate - combine and organize data
- Transform - filter and transform data
- Buffer – protect data
- Edge Analytics - understand data
- Deliver Data - to multiple destinations



Fledge is architected to enable industrial interoperability, advanced application development, cloud portability and system management.

FLEDGE In Energy
Condition Based Monitoring - Transformers

ORACLE

ERP
Trouble Ticketing

DIANOMIC

System Management
Commercial Support

OSIsoft.

T&D Management

kafka

IIOT Pub-Sub Engine

Red Hat

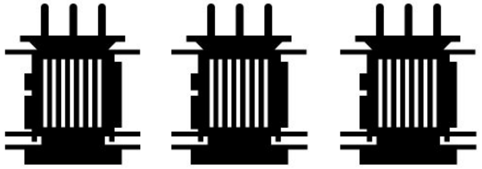
FLEDGE

- Data Collection & Aggregation
- Edge Analytics
- Alerting
- IT-OT System Integration

Ethernet to Cisco 4000

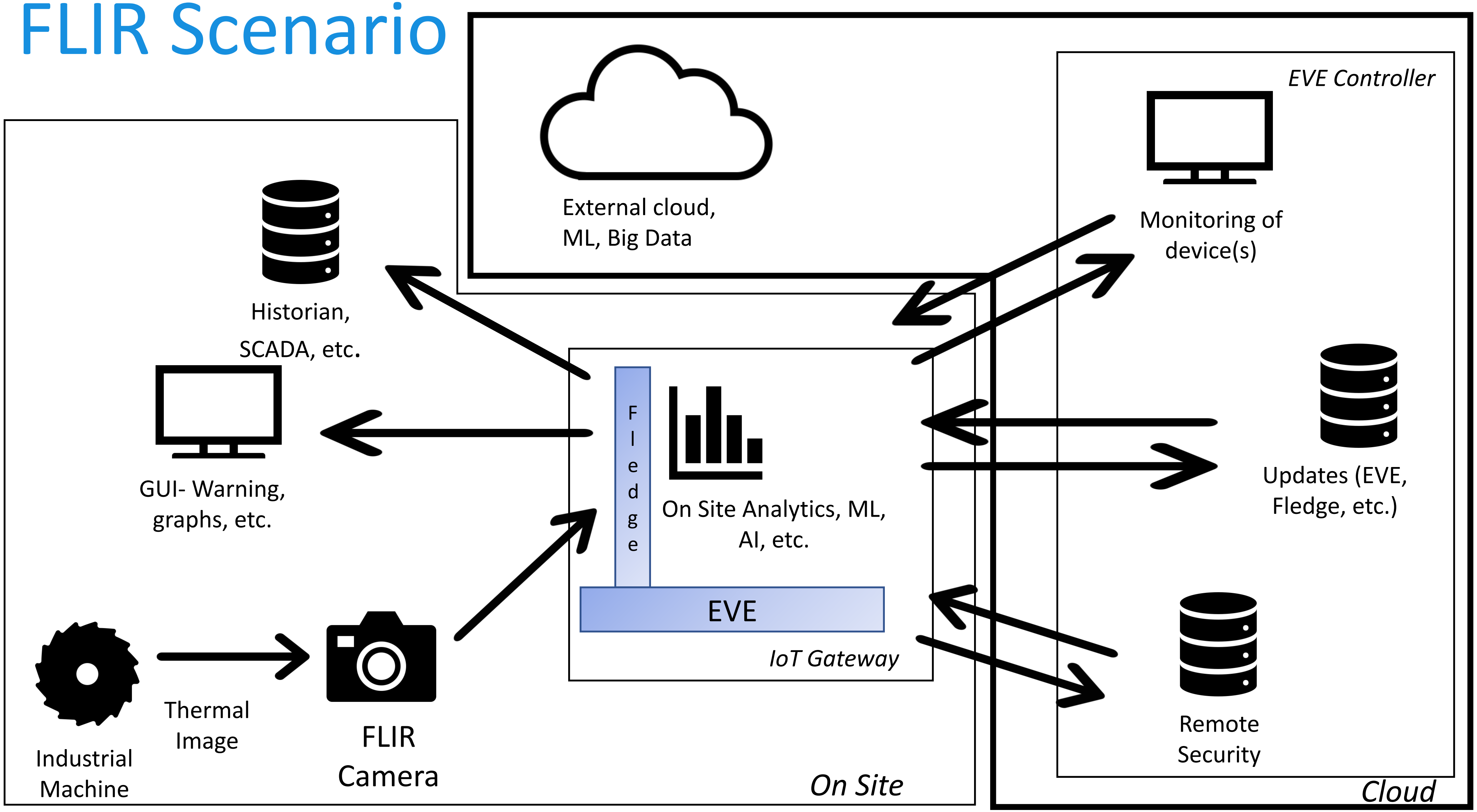
FLIR

FLIR A310
High-Low-Avg Temp
Per Object in Substation



- Monitors substation
- High-Low-Avg Temp Any Object
- Security
- Safety zones
- Safety policy

FLIR Scenario



DEMO

CI/CD

```
make build
./eden --config flir clean
./eden config delete flir
./eden config add flir --devmodel general
./eden config set flir --key adam.eve-ip --value 35.227.137.74
./eden --config flir setup
./eden --config flir start --redis-force=false --adam-force=false --eserver-force=false
 ./eden --config flir eve onboard
 make build-tests
./eden --config flir test ./tests/flir -v debug > ../archives/flir.log
```

Next Steps

LF Edge Community Lab