AWS re:Invent

IOT306-R

Designing secure IoT solutions from edge to cloud

Richard Elberger
Principal Solutions Architect - IoT
Amazon Web Services

AWS re:Invent

aws

Agenda

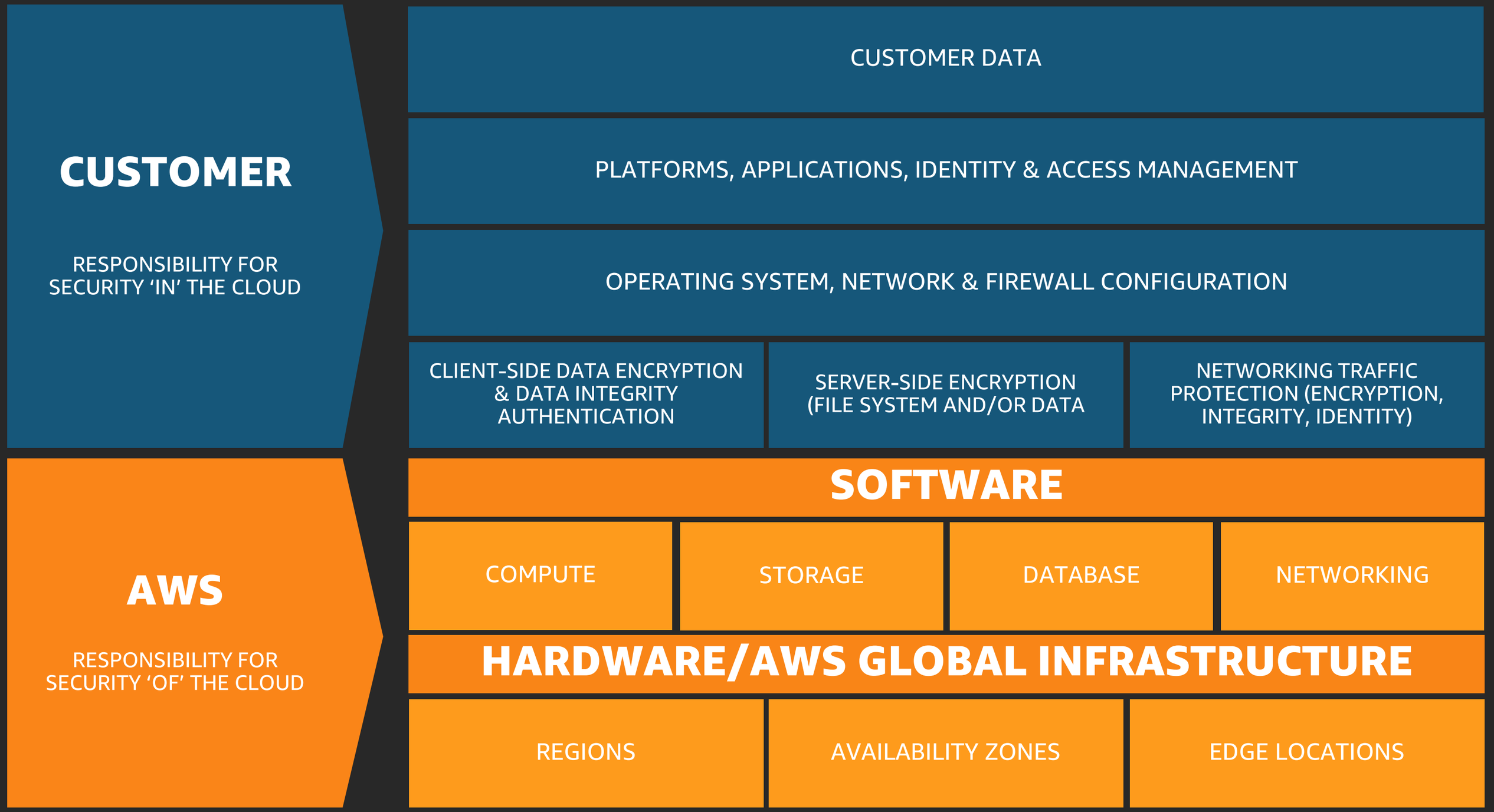
Shared responsibility at the cloud and edge

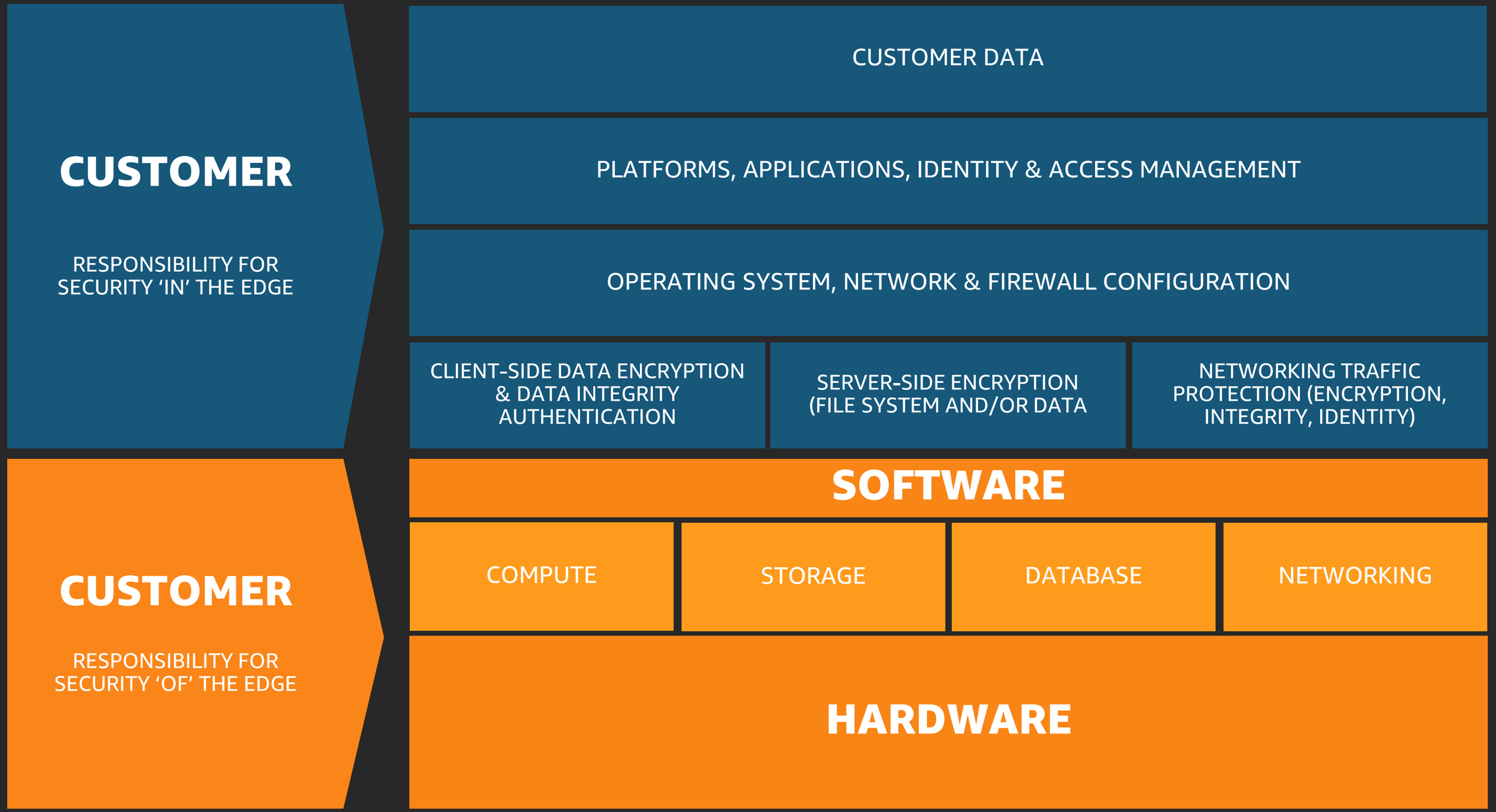
Secure design for industrial IoT

Secure design for smart buildings and cities

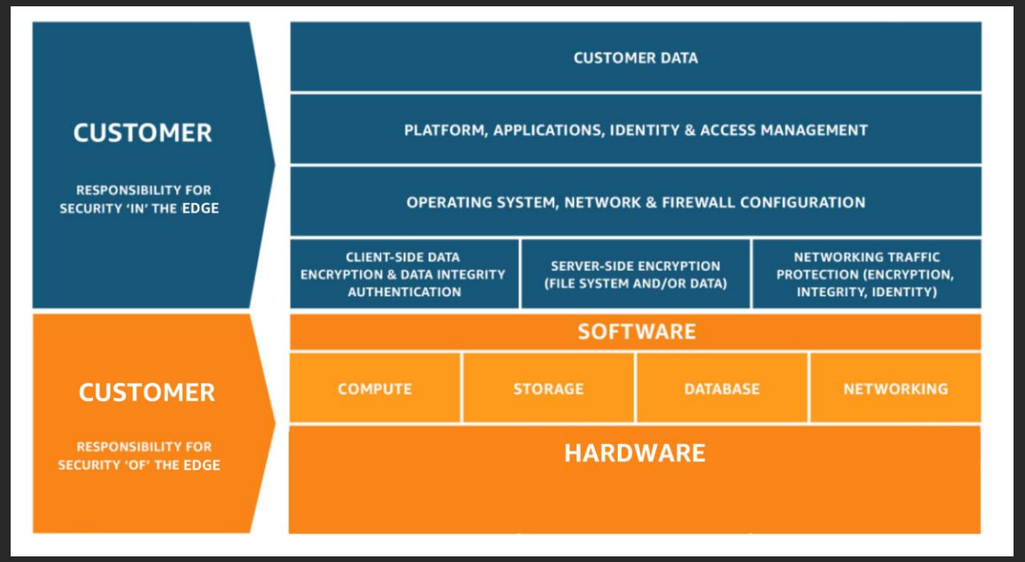
Secure design for consumer IoT

Shared responsibility at the cloud and edge


CUSTOMER
RESPONSIBILITY FOR SECURITY 'IN' THE CLOUD
CUSTOMER DATA
PLATFORMS, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT
OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION
CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA
NETWORKING TRAFFIC PROTECTION (ENCRYPTION, INTEGRITY, IDENTITY)
AWS
RESPONSIBILITY FOR SECURITY 'OF' THE CLOUD
SOFTWARE
COMPUTE
STORAGE
DATABASE
NETWORKING
HARDWARE/AWS GLOBAL INFRASTRUCTURE
REGIONS
AVAILABILITY ZONES
EDGE LOCATIONS

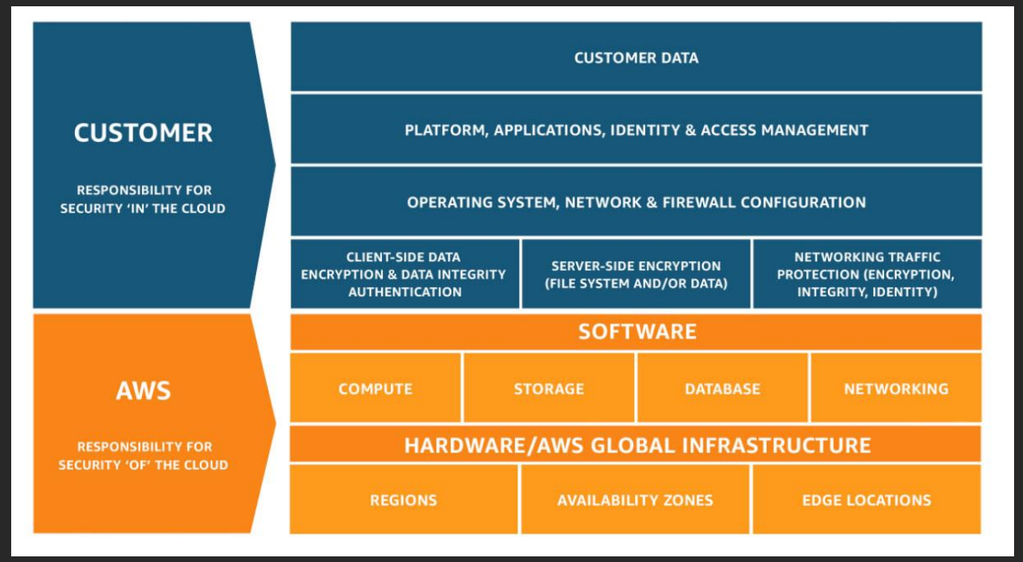

CUSTOMER
RESPONSIBILITY FOR SECURITY 'IN' THE EDGE
CUSTOMER DATA
PLATFORMS, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT
OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION
CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA
NETWORKING TRAFFIC PROTECTION (ENCRYPTION, INTEGRITY, IDENTITY)
CUSTOMER
RESPONSIBILITY FOR SECURITY 'OF' THE EDGE
SOFTWARE
COMPUTE
STORAGE
DATABASE
NETWORKING
HARDWARE

Edge



- Constrained compute
- Constrained energy and storage
- Constrained communications
- Variant environmental conditions

Cloud



- Elastic compute
- Limitless energy and storage
- High bandwidth communications
- Secure physical environment

Threats across the device lifecycle

Genesis

Device authenticity

Black market

Birth

Device identity integrity

Firmware authenticity

Operating

Man in the middle

Brute force

Malware, cryptomining

Distributed Denial of Service (physical, software, communications DDoS)

Personal identity and data theft

Sunset

Backend system still active

Personal identity in device

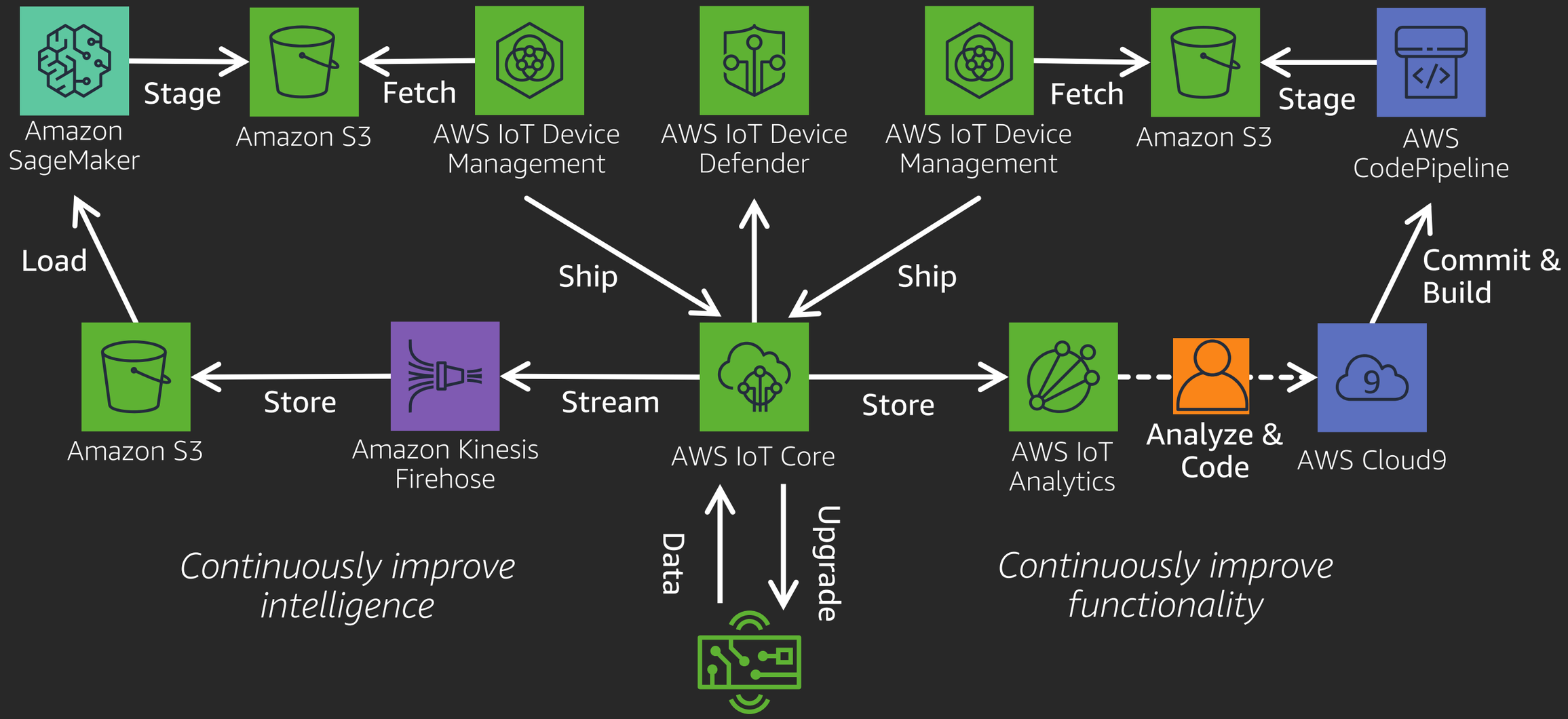
Personal secrets in device

Terminus

Casing reuse

Black market

Use your data for security continuous improvement

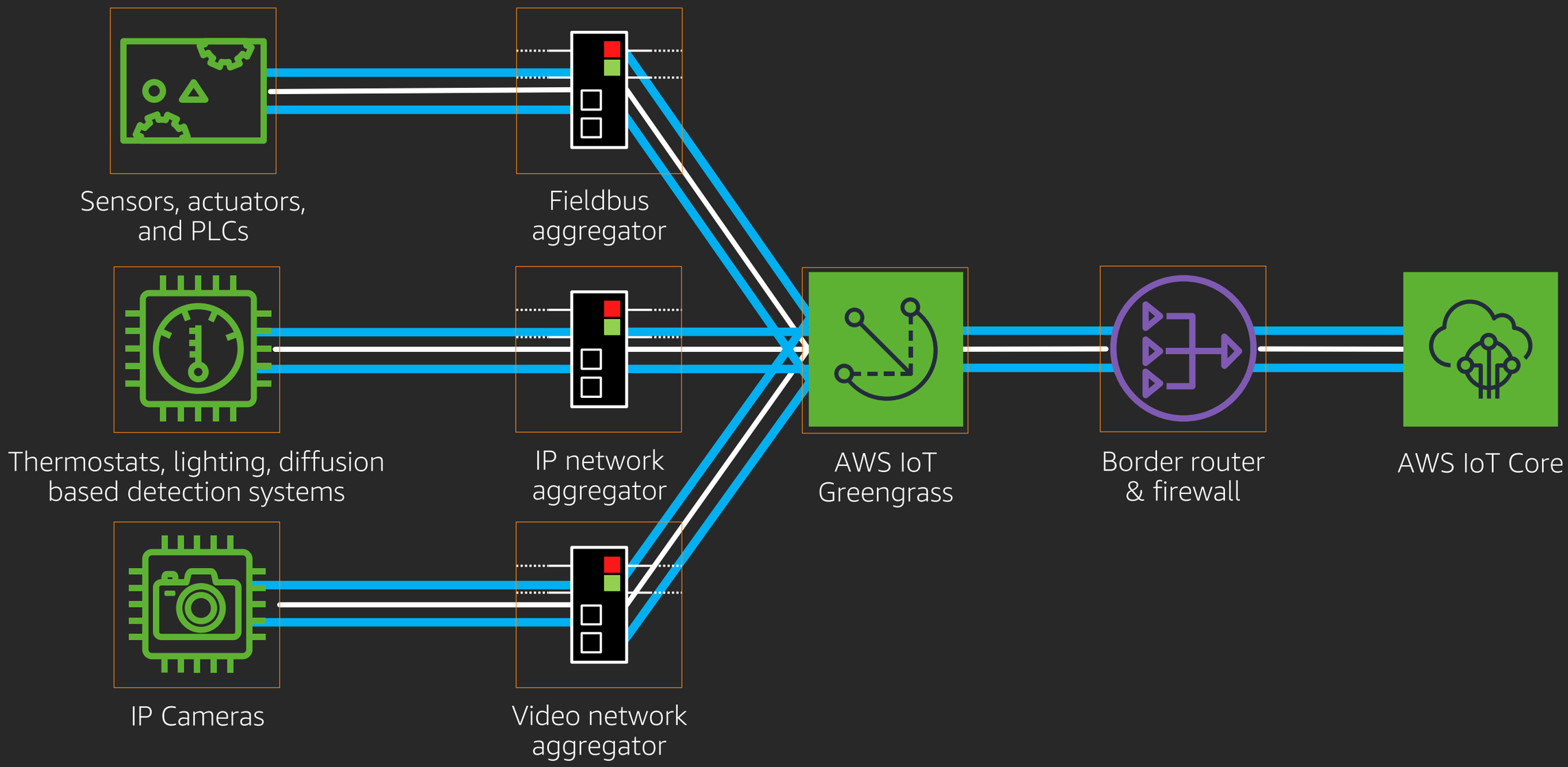


Physical security countermeasures for industrial IoT

AWS re:Invent

aws

Threats to industrial IoT architectures



Identifying industrial IoT physical attacks

invasive	semi-invasive	non-invasive
• Physical destruction • Hardware modification • Tampering	• JTAG • Voltage monitoring • Timing analysis	• Differential power analysis • Differential electromagnetic analysis • Network analysis

Mitigating industrial IoT physical attacks

Tamper protection for enclosures

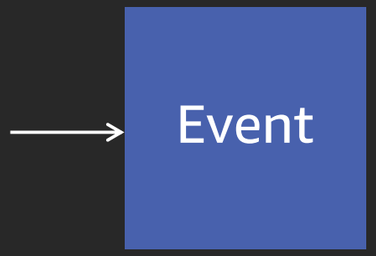
Inductive sensing | Event handler | Message | Actuate

Capture the event, send to AWS IoT Greengrass, identify if someone should be working on the device, and alarm if intrusion detected.

Mitigating industrial IoT physical attacks

Tamper protection for enclosures

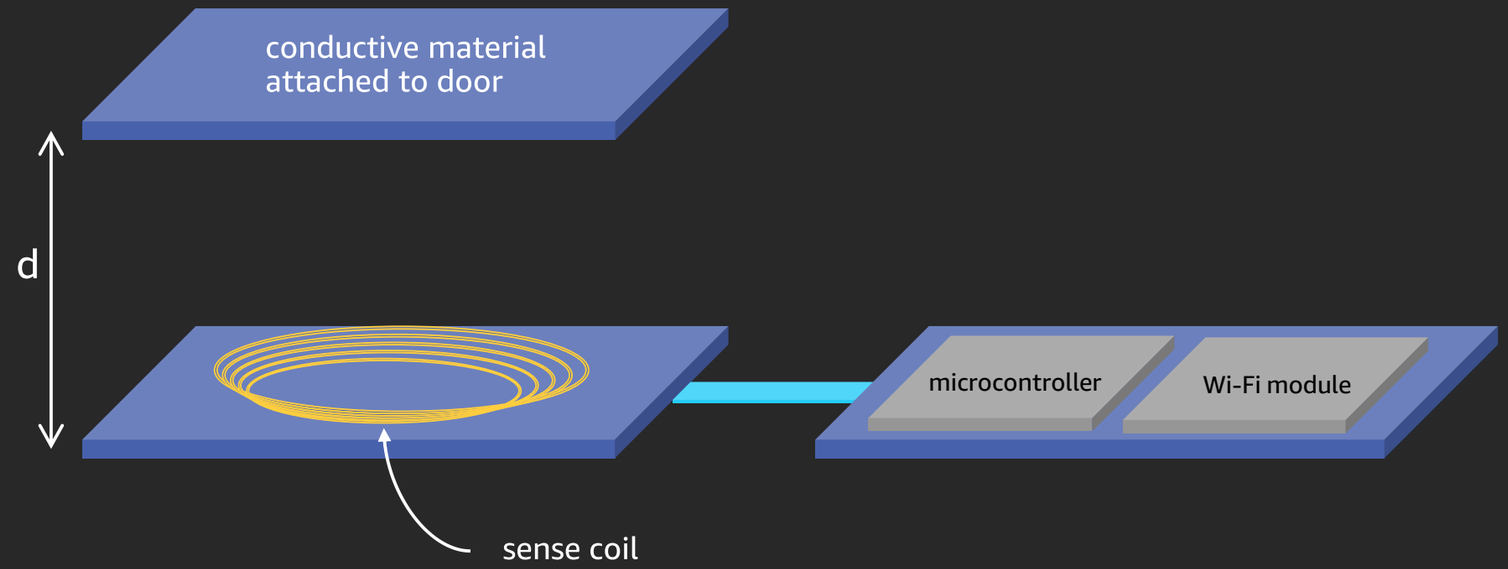
Inductive sensing



Event handler

Message

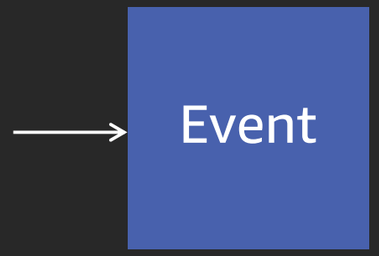
Actuate



Mitigating industrial IoT physical attacks

tamper protection for enclosures

Inductive sensing



Event handler

Message

Actuate

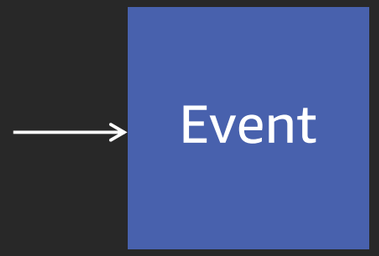
```
typedef struct DoorStatus_t
{
    char * value;
} xDoorStatus_t;
xDoorStatus_t doorstatus;

static void xDoorEventHandler( void * pvParameters )
{
    while (1)
    {
        if ( GPIO_read(DOOR_STATE_IN) == 1) // Opened
        {
            if (strncmp(doorstatus.value, "1", 1) == 0) continue;
            configPRINTF(("Door is open\r\n"));
            doorstatus.value = "1";
            GPIO_write(DOOR_OPENED_OUT, 1);
            GPIO_write(DOOR_CLOSED_OUT, 0);
        }
```

```
        else if ( GPIO_read(DOOR_STATE_IN) == 0 )  // Closed
        {
            if (strncmp(doorstatus.value, "0", 1) == 0) continue;
            configPRINTF(("Door is closed\r\n"));
            doorstatus.value = "0";
            GPIO_write(DOOR_OPENED_OUT, 0);
            GPIO_write(DOOR_CLOSED_OUT, 1);
        }

        xQueueSend( xDoorNotifyQueue, &doorstatus, pdMS_TO_TICKS( 500
) );

        vTaskDelay( pdMS_TO_TICKS( 100 ) );
    }
}
```

Mitigating industrial IoT physical attacks

Tamper protection for enclosures

Inductive sensing



Event handler

Message

Actuate

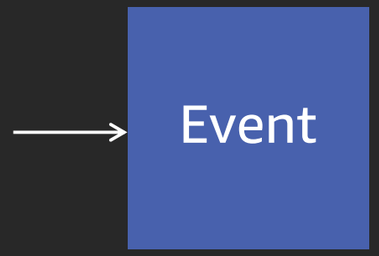
```c

static void prvMqttTempTask( void * pvParameters )
{
  ...
  while (1)
  {
    if( xQueueReceive( xInductiveSensorNotifyQueue,
                       &queueMessage,
                       sensorRECV_QUEUE_WAIT_TICKS == pdFAIL )
continue;

   snprintf( cDataBuffer, sizeof( cDataBuffer),
             "{\”alarm\":%d, \”boxId\”:%d}",
             queueMessage.reading, boxId );
```

```
MQTTAgentPublishParams_t pxPublishParams;

pxPublishParams.pucTopic =
  ( uint8_t * ) pxParameters->topic;

pxPublishParams.usTopicLength =
  ( uint16_t )  strlen( ( const char * ) pxParameters->topic );

pxPublishParams.pvData = cDataBuffer;

pxPublishParams.ulDataLength =
  ( uint32_t )  strlen( ( const char * ) cDataBuffer );

pxPublishParams.xQoS =  eMQTTQoS0;

If ( MQTT_AGENT_Publish( xMQTTHandle, &( pxPublishParams ),
                         sensorMQTT_TIMEOUT ) != eMQTTSuccess )
       { ... Put retry, local data logging / alert logic here ... }
}
```

Mitigating industrial IoT physical attacks

Tamper protection for enclosures

Inductive sensing

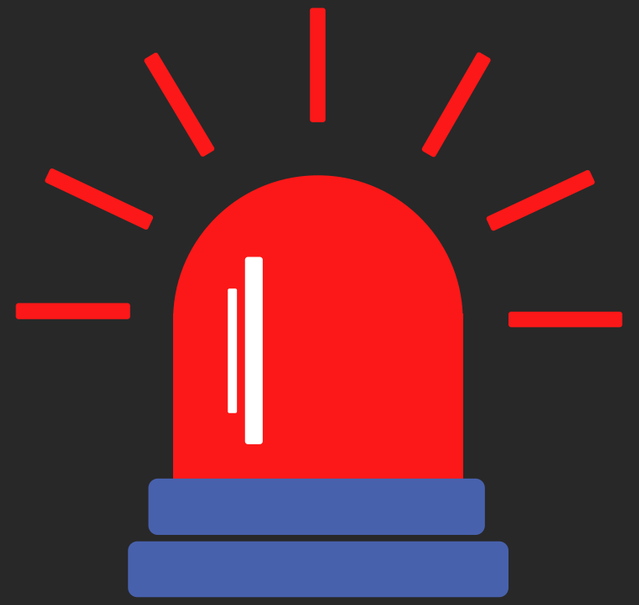


Event handler

Message

Actuate

```
channel_output_red   = 38 # /sys/class/gpio/gpio38
channel_output_green = 36 # /sys/class/gpio/gpio36

GPIO.setmode(GPIO.BOARD)
# Output for the door open event
GPIO.setup( channel_output_red,   GPIO.OUT ) # Door opened
GPIO.setup( channel_output_green, GPIO.OUT ) # Door closed

def lambda_handler(event, context):
    # Identify the state.
    if event['state']['desired']['door_state'] == 1:
        GPIO.output(channel_output_red, GPIO.HIGH)
        GPIO.output(channel_output_green, GPIO.LOW)
    else:
        GPIO.output(channel_output_red, GPIO.LOW)
        GPIO.output(channel_output_green, GPIO.HIGH)
```

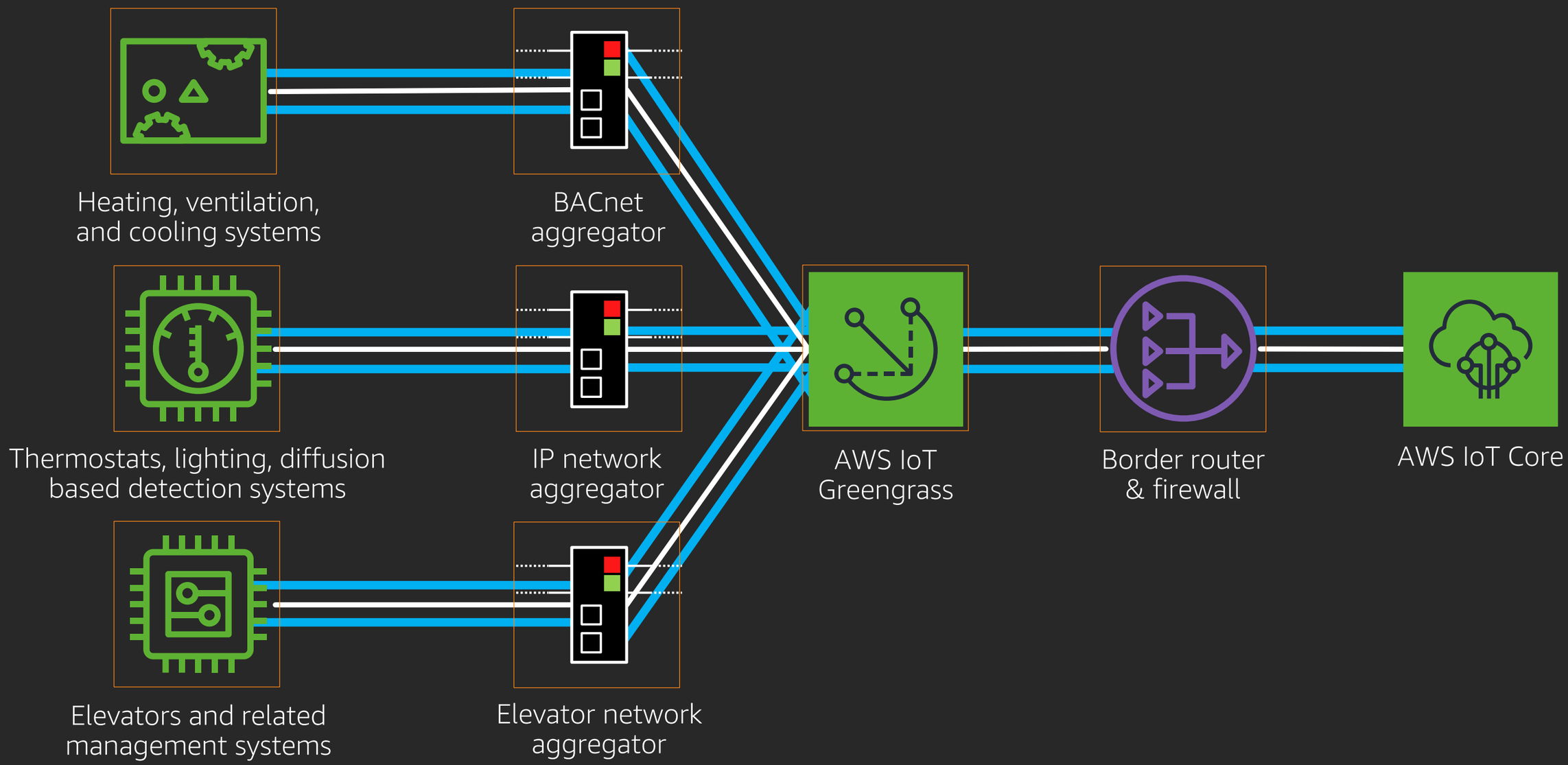


Software security countermeasures for smart buildings and cities

AWS re:Invent

aws

Threats to smart building architectures

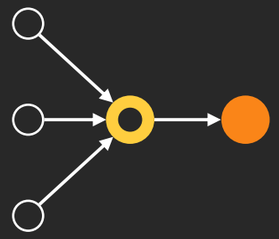


Identifying smart building software attacks

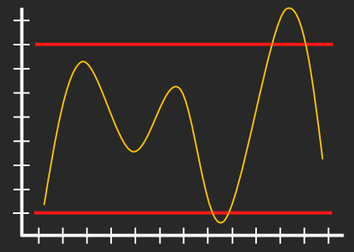
invasive	semi-invasive	non-invasive
• Firmware replacement • Application replacement • Application permanent DDOS / bootloader corruption	• Malware / ransomware • Cryptocurrency mining • Masquerading / botnets • Application DDOS	• Network analysis • Network hijacking

Mitigating smart building software attacks

Application and operating system updates



Gather and aggregate



Evaluate statistics

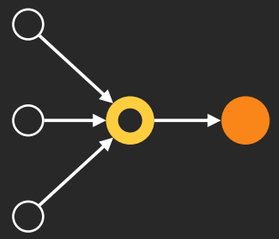
Fix and build

Deploy to fleet

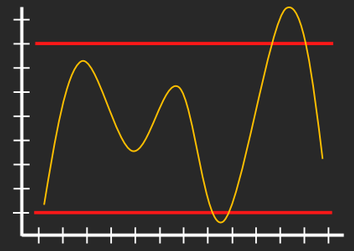
Gather and aggregate system metrics, evaluate out of bounds statistics to determine attack surface vulnerability, fix and build, deploy to fleet

Mitigating smart building software attacks

Application and operating system updates



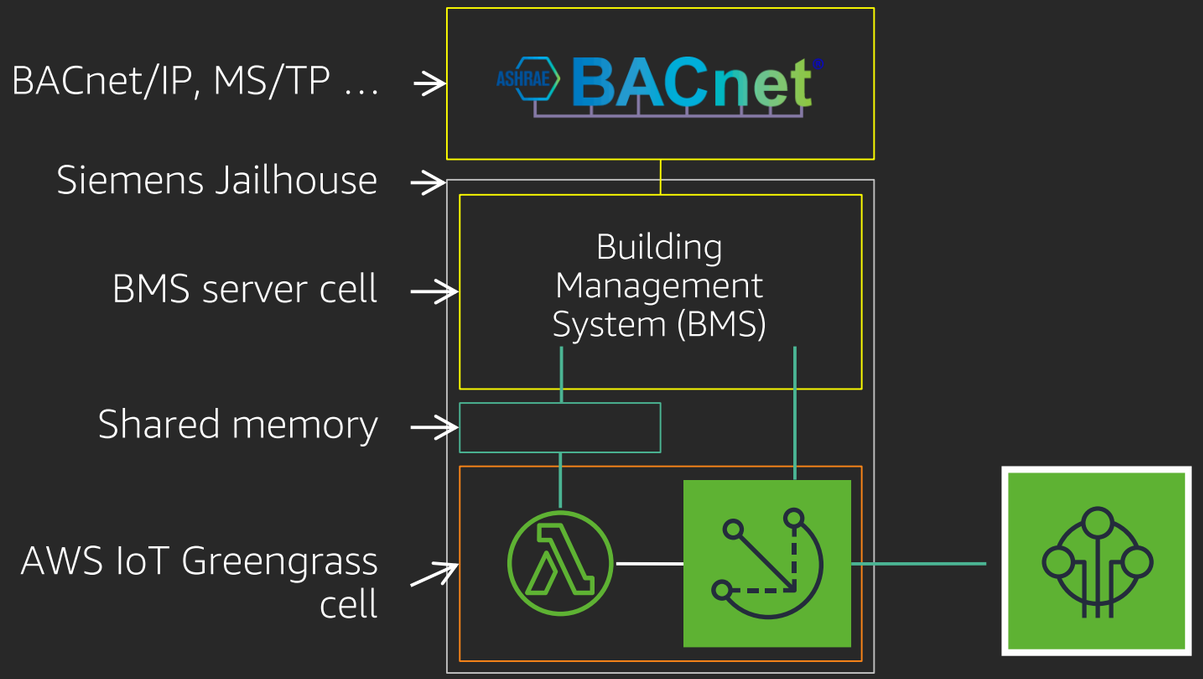
Gather and aggregate



Evaluate statistics

Fix and build

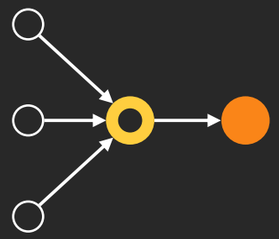
Deploy to fleet



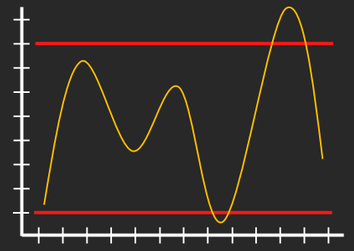

```
... other routes ...

{ "Id":      ”BACnetListenerToCloudRoute",
  "Source":  ”BACnetListener",
  "Subject": ”building/10/bacnet/in",
  "Target":  ”Cloud” },

{ "Id":      ”CloudToBldgMgrHandlerRoute",
  "Source":  ”GGShadowService",
  "Subject": ” $aws/things/bldg10mgr/shadow/update/delta",
  "Target":  ”BldgMgrHandler" },

{ "Id":      ”BldgMgrHandlerToBACnetProviderRoute",
  "Source":  ”BldgMgrHandler",
  "Subject": “building/10/bacnet/out",
  "Target":  ”BACnetProvider" },

... other routes ...
```

Mitigating smart building software attacks

Application and operating system updates



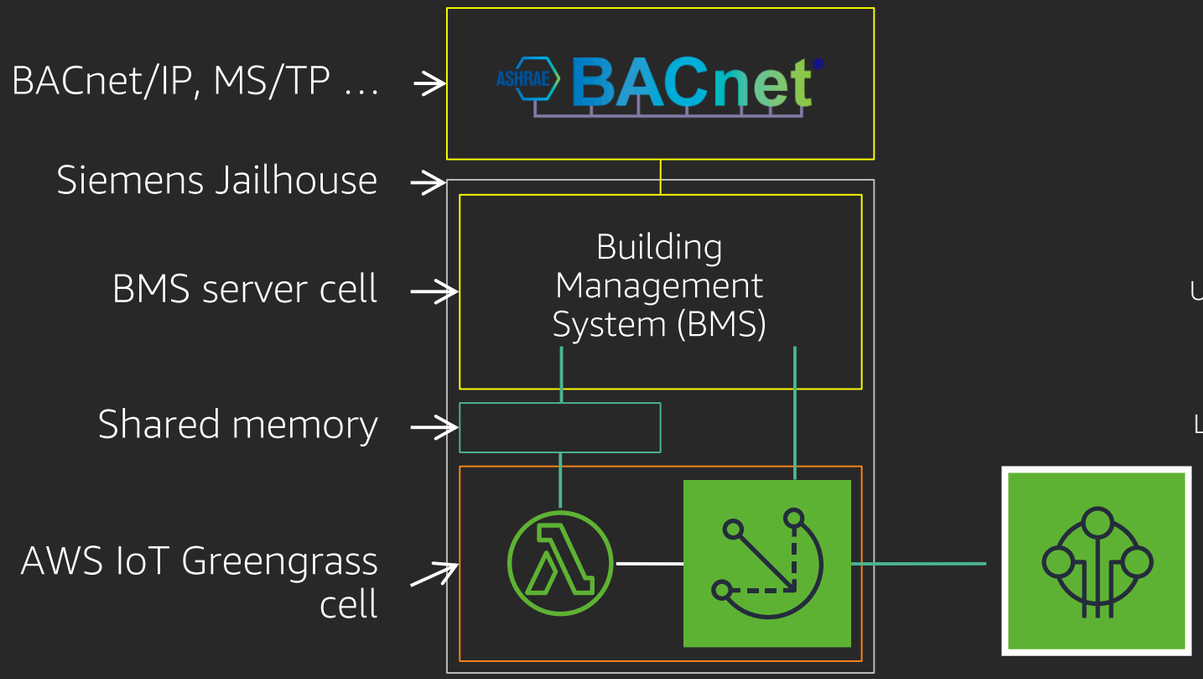
Gather and aggregate

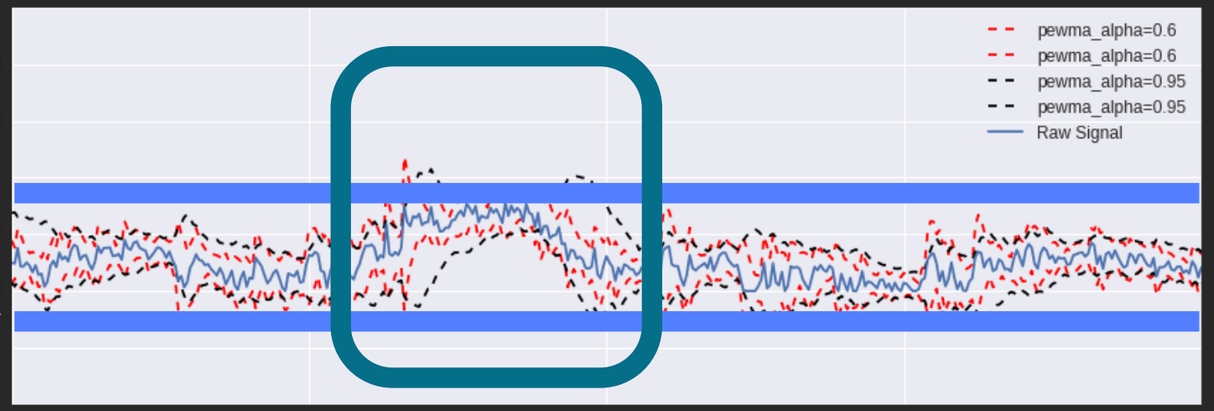


Evaluate statistics

Fix and build

Deploy to fleet

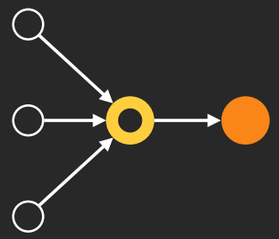




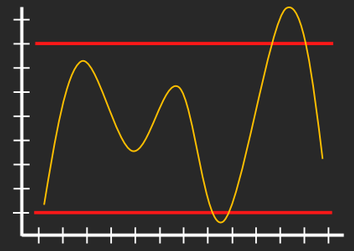
https://aws.amazon.com/blogs/iot/anomaly-detection-using-aws-iot-and-aws-lambda/

Mitigating smart building software attacks

Application and operating system updates



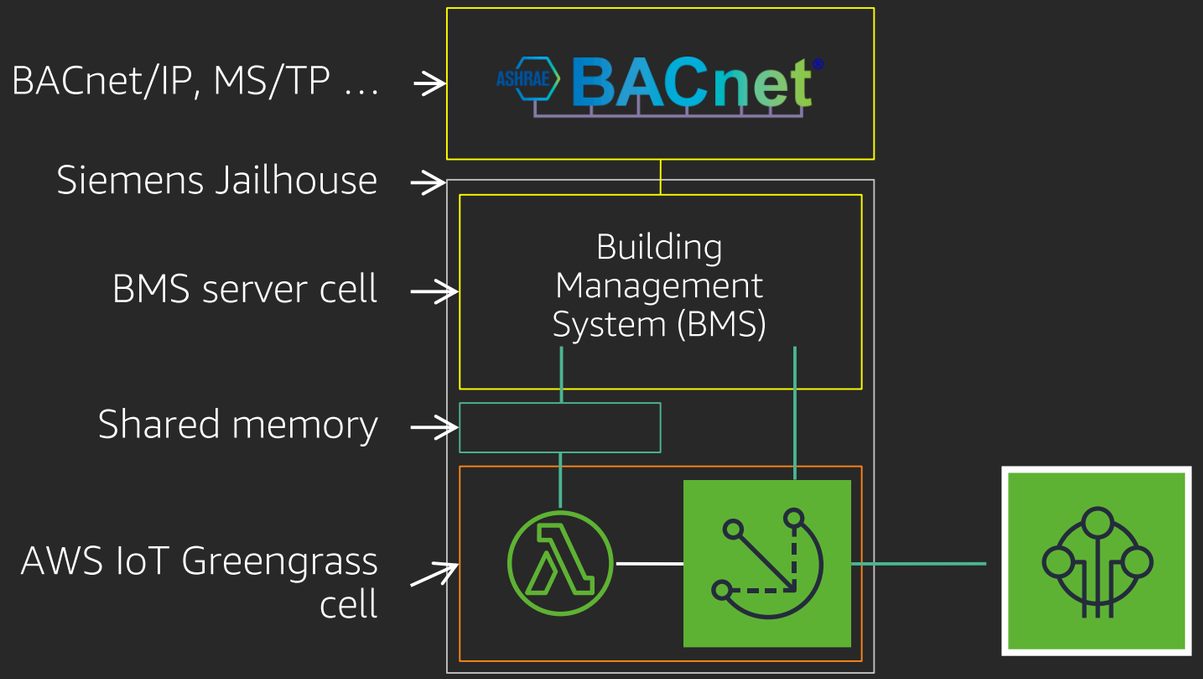
Gather and aggregate



Evaluate statistics

Fix and build

Deploy to fleet



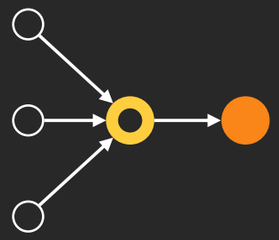

```bash
object_version=$(aws s3api put-object                    \
                      --bucket ${SOURCE_S3_BUCKET}       \
                      --key ${target_file}               \
                      --body ../bin/${target_file}       \
                      --query VersionId --output text)
```

```
SOURCE_CONFIG="s3={bucketName=${SOURCE_S3_BUCKET},key=${target_file},
version=${object_version}}"
TARGET_CONFIG="s3={bucketName=${TARGET_S3_BUCKET}}"
```

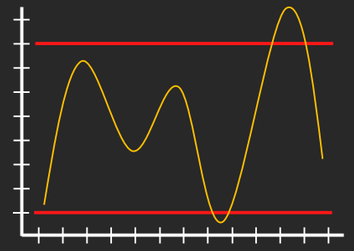
```
signer_jobid=$(aws signer start-signing-job        \
                     --source       ${SOURCE_CONFIG} \
                     --destination  ${TARGET_CONFIG} \
                     --profile-name bnd3_1025_200955 \
                     --query jobId --output text)
```

Mitigating smart building software attacks

Application and operating system updates



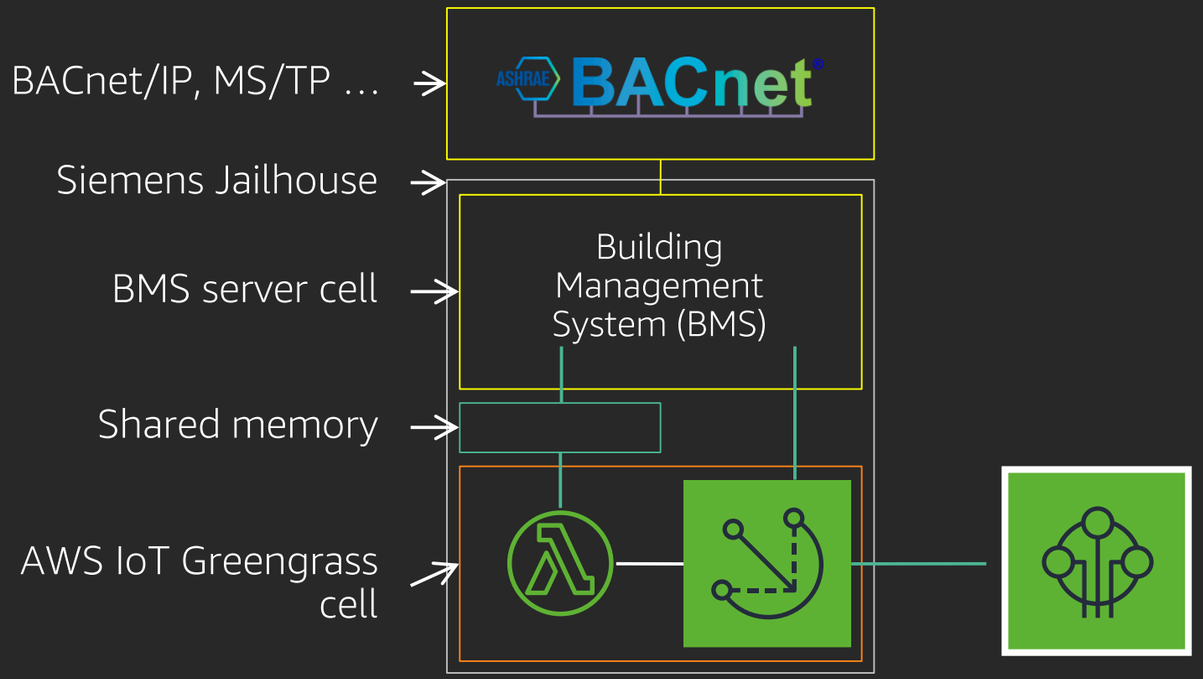
Gather and aggregate



Evaluate statistics

Fix and build

Deploy to fleet




```bash
JOBID=$(uuidgen)

aws iot create-job --output text                                \
  --job-id ${JOBID}                                             \
  --targets ControlPanels                                       \
  --document file://upgrade-control-panels.json                 \
  --target-selection SNAPSHOT                                   \
  --query jobArn
```

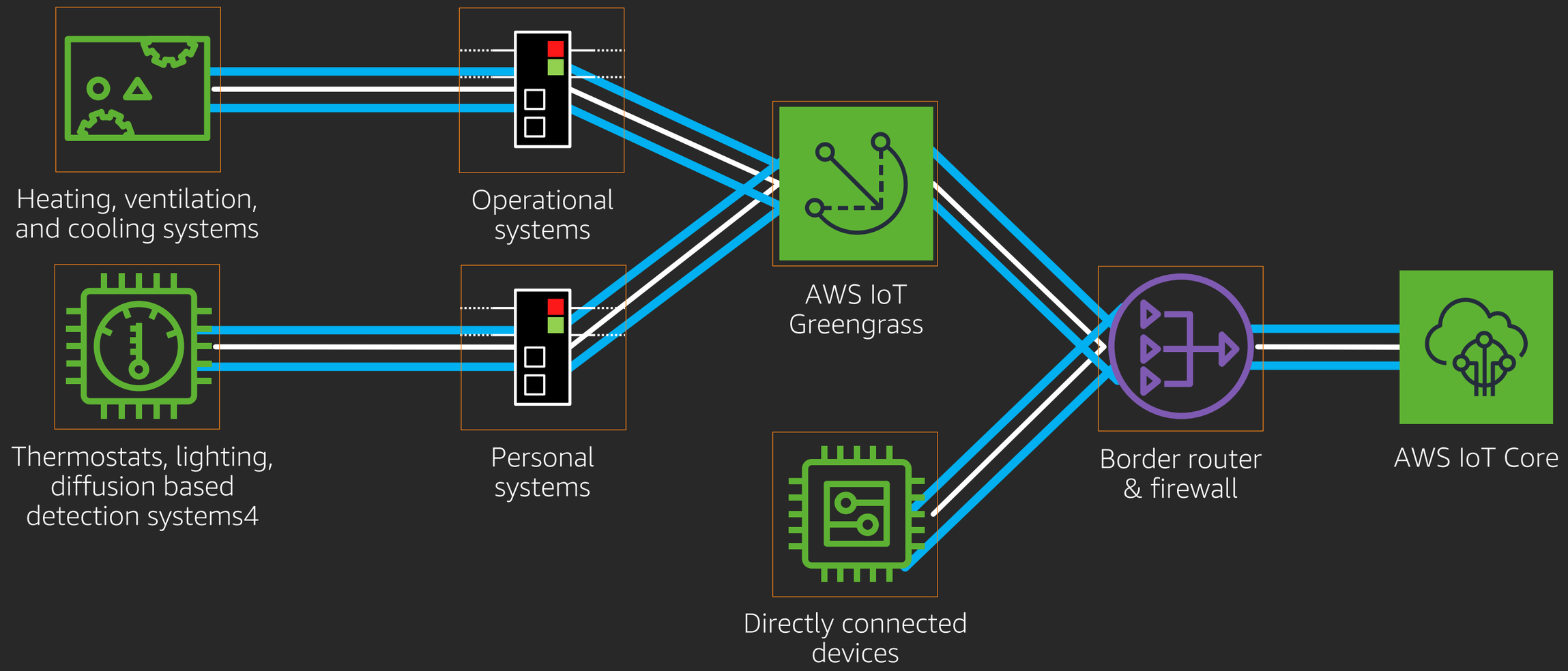
```
STATUS=IN_PROGRESS
while x${STATUS} != xIN_PROGRESS; do
  STATUS=$(aws iot describe-job --output text                   \
                --job-id ${JOBID}                               \
                --query job.status)
  sleep 3
```

Communication security countermeasures for consumer IoT

AWS re:Invent

aws

Threats to connected home architectures



Identifying connected home communication attacks

invasive	semi-invasive	non-invasive
• Network hijacking • Device hijacking • Device identity theft	• Man in the middle • Malware/Ransomware (routers) • Botnets (routers)	• Network analysis • Radio interference

Ways to move from device genesis to birth

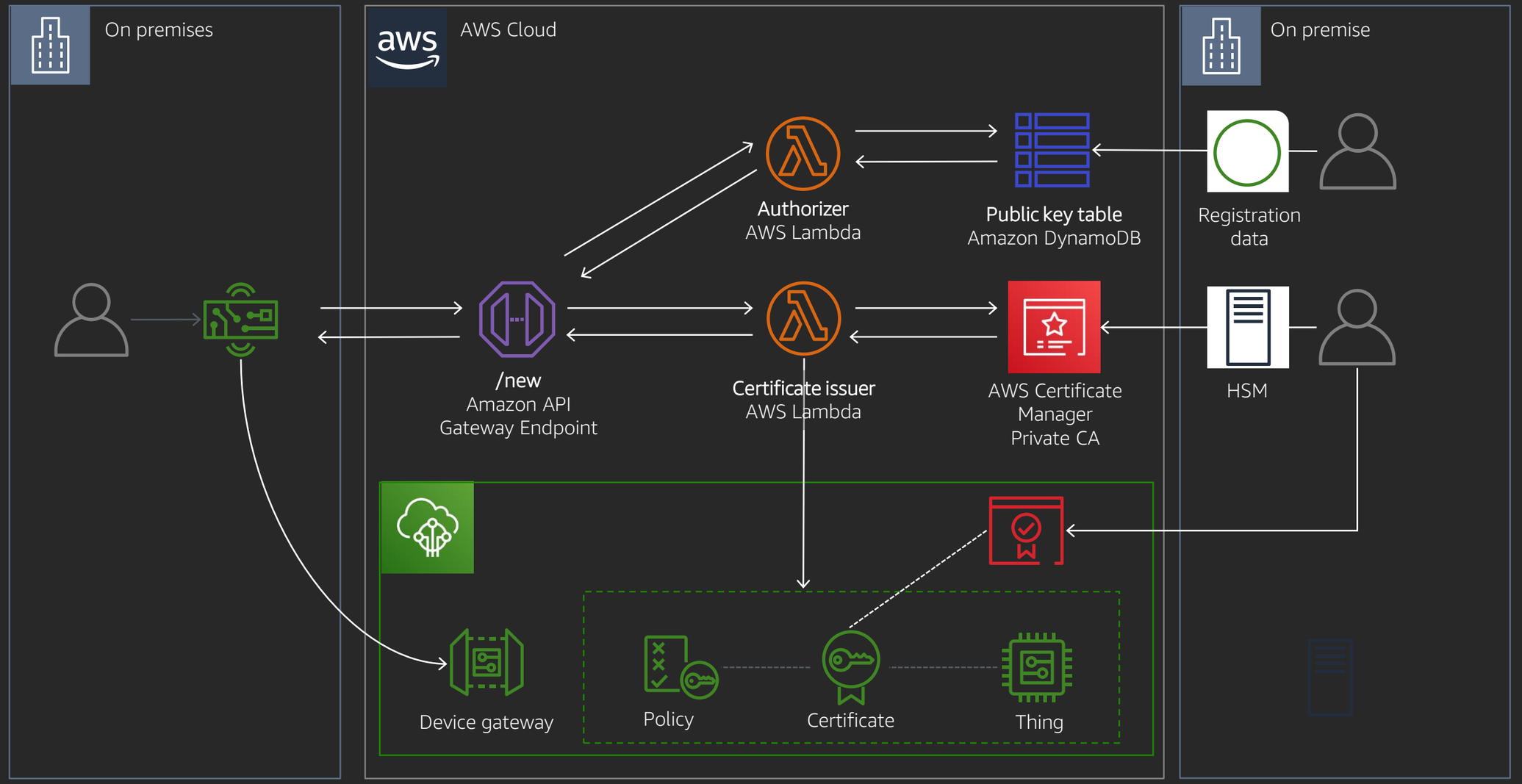
CM injects private key and certificate to flash

Secure element or enclave vendor injects certificate

Provision using a birthing certificate service

Provision using a Certificate Signing Request on demand

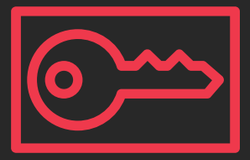
Provisioning at runtime with secretfree



https://github.com/aws-samples/iot-provisioning-secretfree

Mitigating connected home communication attacks

Secure device identity provisioning for pristine connectivity



Device identity

Device whitelisting

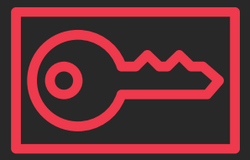
Provisioning

Connect

Ensure device identity privacy, whitelist authentic hardware, decouple credential provisioning, and ensure transport layer security.

Mitigating connected home communication attacks

Secure device authentication and authorization for pristine connectivity



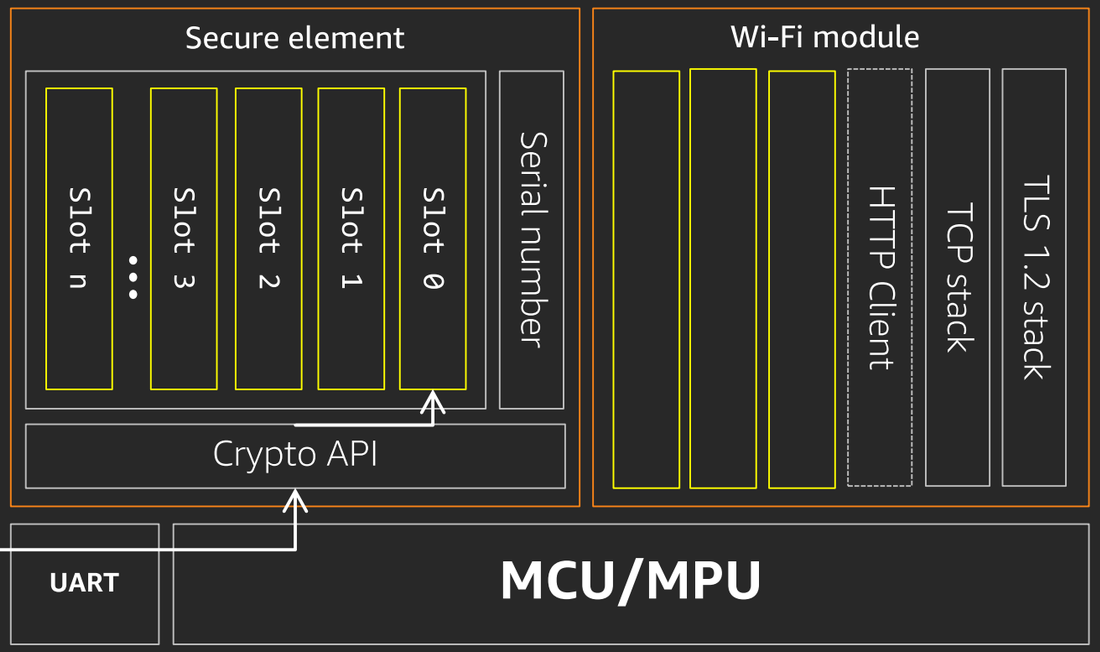
Device identity

Device whitelisting

Provisioning

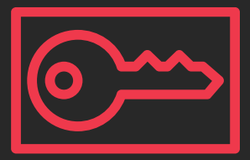
Connect

```c
... Device initialization

status = initATCADevice(&discoverCfg, &disc_device);

status = atcab_genkey(slot, *public_key);
```



Mitigating connected home communication attacks

Secure device authentication and authorization for pristine connectivity



Device identity

Device whitelisting

Provisioning

Connect

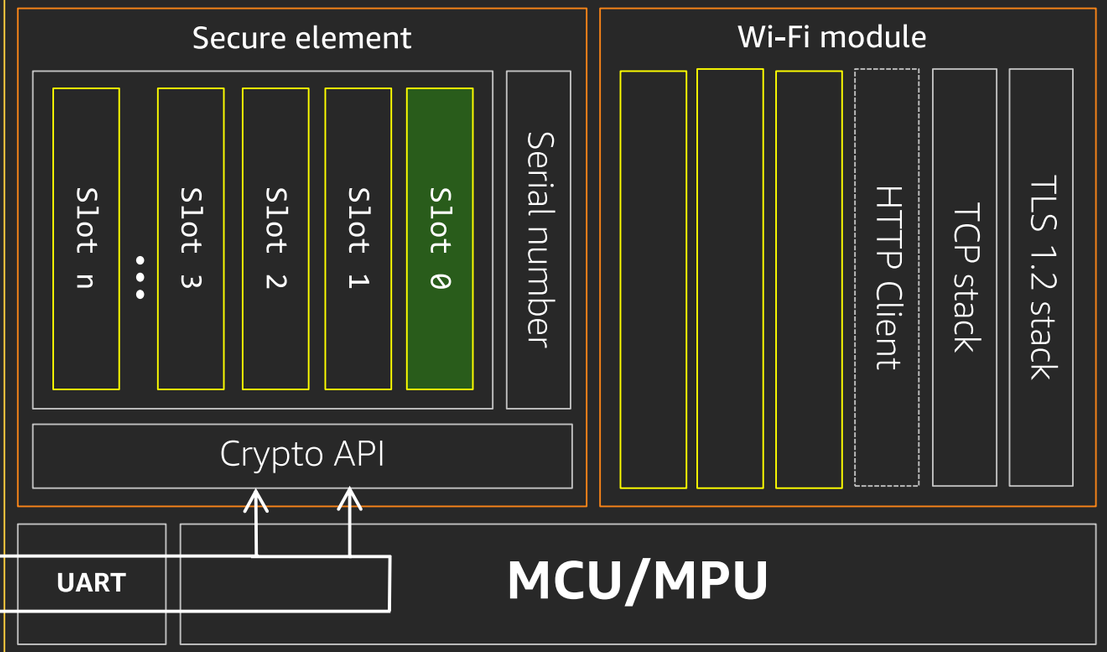
```
// Retrieves device serial number
atcab_read_serial_number( *serial_number );
```

```
// Retrieves public key
atcacert_read_device_loc( device_loc, data);
```

```
// Emit configuration over serial...
```

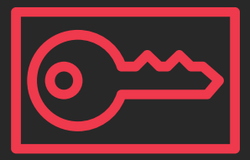
```text
a93184c377e4cfddc4a485332b2df151 LS0tLS1CRUdJTiBQ...tFWS0tLS0tCg==
abe185a972e12e79f508b0ef2a2df151 LS0tLS1CRUdJTiBQ...tFWS0tLS0tCgo=
```

```bash
aws dynamodb put-item --table-name provisioning-db \
 --item "{\"device-id\": {\"S\": \"$serial\"}, \"pubkey\": {\"S\":\"$pubkey\"}}"
```



Mitigating connected home communication attacks

Secure device authentication and authorization for pristine connectivity



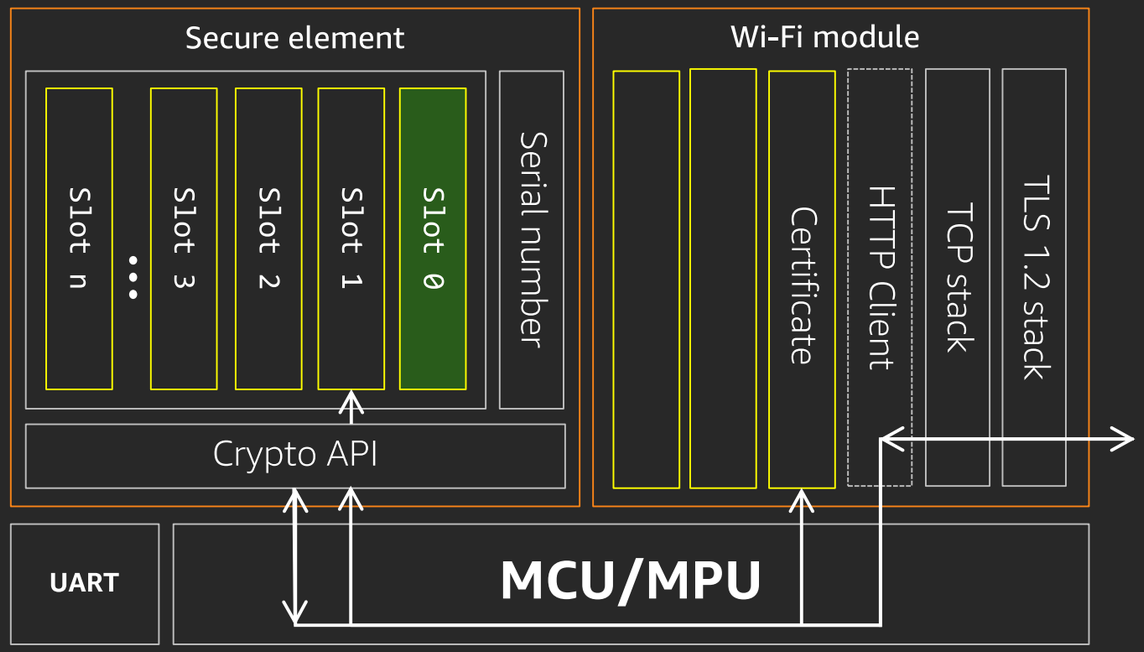
Device identity

Device whitelisting

Provisioning

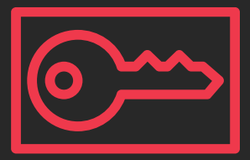
Connect

```c
xResult = atcacert_create_csr_pem( csr_def,
                                   csr,
                                   csr_size);
...

xResult = IotHttpsClient_SendSync( xHTTPConnection,
                                   xHTTPRequest,
                                   &xHTTPResponse,
                                   &xHTTPResponseInfo,
                                   provHTTP_TIMEOUT_MS );
configASSERT( xHTTPClientResult == IOT_HTTPS_OK );
...
xResult = atcacert_write_cert( cert_def,
                               cert,
                               cert_size);
```



Mitigating connected home communication attacks

Secure device authentication and authorization for pristine connectivity



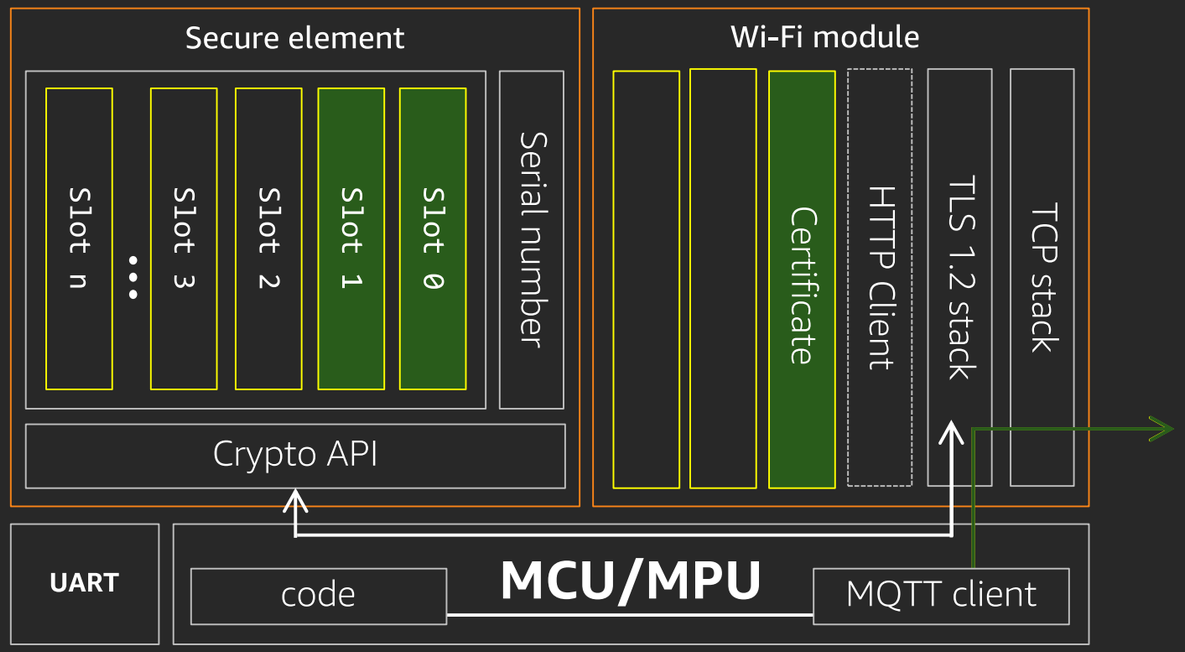
Device identity

Device whitelisting

Provisioning

Connect

```c
static IotMqttConnection_t * pMQTTHandle;
...
IotMqttNetworkInfo_t networkInfo = IOT_MQTT_NETWORK_INFO_INITIALIZER;
...
IotMqttConnectInfo_t connectInfo = IOT_MQTT_CONNECT_INFO_INITIALIZER;
...
connectStatus = IotMqtt_Connect( &networkInfo,
                                 &connectInfo,
                                 MQTT_TIMEOUT_MS,
                                 pHandle );
```



Recap

What we have learned

Connectivity security patterns reduce basic threats

Physical security is required to protect device identity

Device analytics mitigates ongoing operational threats

Device management eases over-the-air firmware and config updates

Threats and countermeasures are always evolving

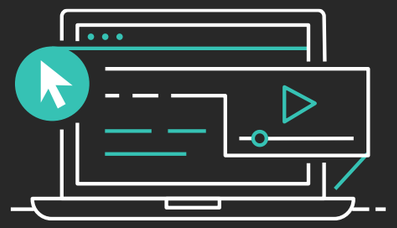
All demos will be released to Github after re:Invent!

Secure connectivity demo **DEM127** - Wednesday, Dec 4, 5:30 PM - 5:50 PM

Learn IoT with AWS Training and Certification

Resources created by the experts at AWS to help you build IoT skills

Take the free digital curriculum, Internet of Things (IoT) Foundation Series, to build IoT skills and work through common scenarios



25+ additional free digital courses cover topics related to IoT, including:

- AWS IoT Core
- AWS IoT Greengrass
- AWS IoT Analytics
- AWS IoT Device Management
- AWS IoT Events

Visit the Learning Library at https://aws.training

aws training and certification

Thank you!

Richard Elberger

Github rpcme

LinkedIn https://www.linkedin.com/in/richardelberger/

Twitter richardelberger

Please complete the session survey in the mobile app.