asvin

SECURE UPDATES
in Internet of Things

asvin.io

TECHNICAL WHITEPAPER

Table of Contents

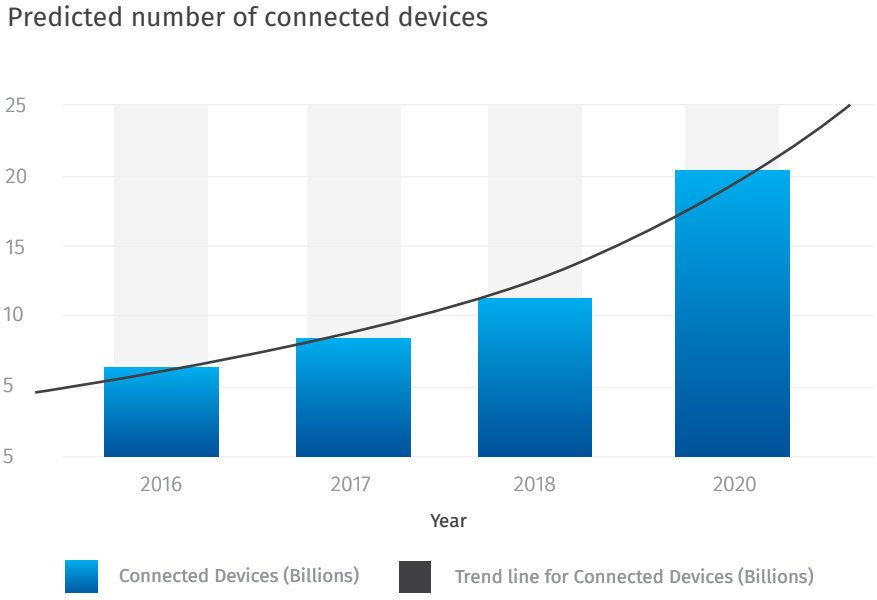
Mission

"As'vin" (Sanskrit) is the name of the divine twins, the healer of gods in Hindu mythology. The spirit of these twins is embedded into the core mission of asvin.io:

"Healing the Internet of Things from security flaws and vulnerabilities by providing an easy and blockchain secured update distribution service."

asvin.io provides secure update solution among parties of Internet of Things applications: for the embedded device controller (edge devices) and server side device cloud applications. Today unpatched vulnerabilities on IoT devices are door opener for diseases as DDoS attacks, Ransomware, Hacking, and Data theft. Updating insecure IoT systems is analogous to applying medicines to sick patients. There is a growing need for continuously healing infected or unsecure devices in the Internet of Things. That's why we call our solution asvin.io. IoT devices have become an important part of our technological and social ecosystem. The number of IoT devices in operation are growing exponentially. There are billions of these devices installed in our homes and workplaces. Humans are surrounded by sensors, actuators and routers, which constantly monitor external events, react to them by taking a logical decision, and transmit notifications to users. To put this into perspective Gartner estimates that there are 8.4 billion connected devices in 2017 and predicts that this will rise to 20.4 billion devices by 2020[1] , that is 3 times the world's human population.

1 https://stormagic.com/welcome-to-the-edge-the-need-for-iot-edge-computing/



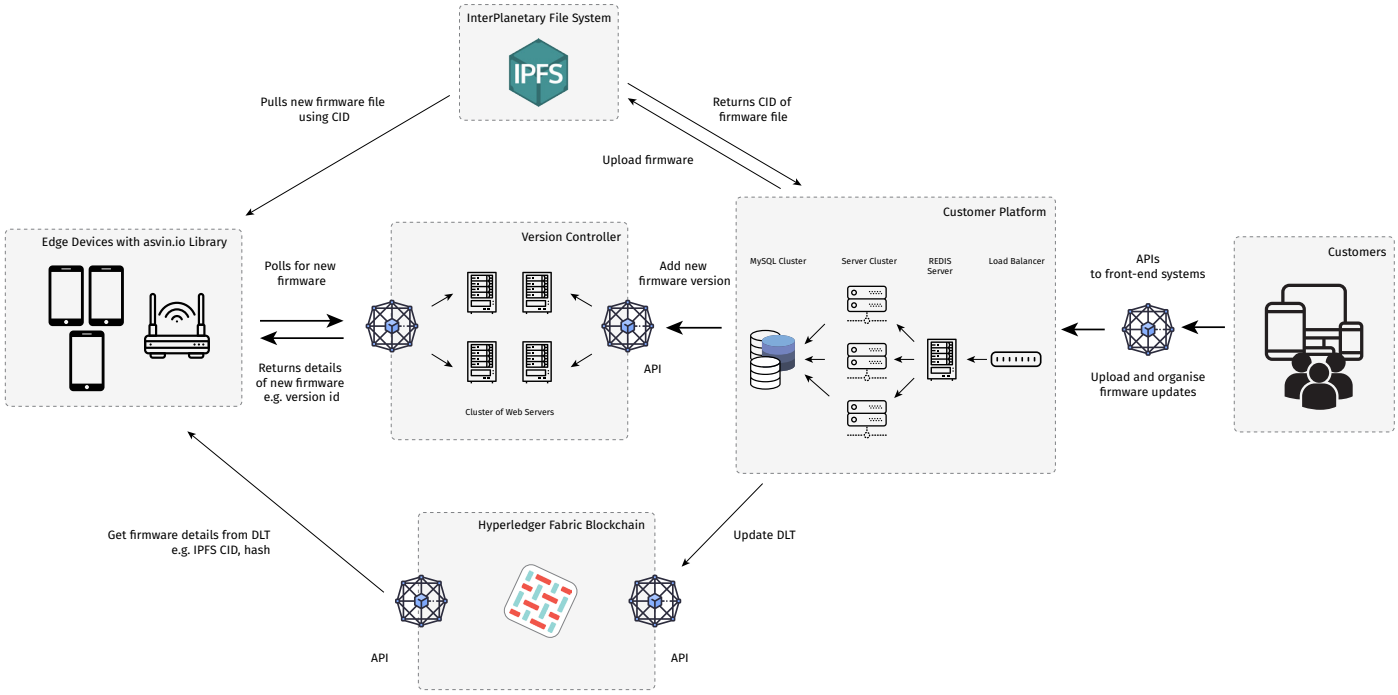
These devices are safety critical and failure of these devices might result in tremendous financial loss and could incur in loss of human life too. For an instance, take a smoke detector installed in our house. In Stuttgart there are 600,000 inhabitants and considering workplaces there should be more than 1 million smoke detectors in place to keep us safe. In a fast growing technology oriented ecosystem there might be a need to update the network of these devices for any reason like, to make the network more robust, government laws and policies etc. In such a scenario firmware of these devices needs to be updated to cope up with the changes in the network. It is not feasible to update firmware of millions of devices manually. asvin.io provides a platform to update the devices over the air, secured by blockchain technology.

Open Source License Policy

At asvin.io., we believe in Open Source technologies. It is our conviction that transparency and freedom of usage are important to build trusted, sustainable and sage solutions in Internet of Things. By that, we license our own developed components under Apache 2.0 license, which is “real open source” accepted by the free software foundation.

General Architecture

The staggering number of IoT edge devices constantly generate enormous amounts of data. The traditional centralized platforms cannot perform the device management and firmware roll out management on a strict timeline basis, more often a quick response is needed, for an instance roll out a new firmware for all baby monitors in Berlin in case of malicious attack. To prevail over the network latency, security and bandwidth limitations of the legacy computation platforms, asvin.io employs distributed cloud computing technology to accommodate and manage billions of devices, store their firmware on server, and responds to device requests on strict timely manner. asvin.io is a cloud-based solution which provides a single point control to its customer over all IoT devices deployed in the market. The following figure depicts asvin.io platform.



The asvin.io computing model is comprised of 5 different entities. Each entity on asvin.io platform is tailored to manage a special task. This way, asvin.io ensures scalability and defense against single point of failure.

Edge Devices with asvin.io Library

The edge devices are end points in the asvin.io architecture. These are the end devices in an IoT network which are operated to control, manage and solve a physical task for an instance, smart washing machine in a house, temperature and humidity monitor in a chemical plant, air quatliy sensor installed in a city etc. These devices have microcontrollers and sensors at their core. Usually, many edge devices shall have small footprint and must be easy to manage in remote areas under often extreme environmental conditions. For example: industrial process monitoring sensors, smart meters, Lora node etc.
asvin.io provides libraries for typical edge devices to enhance them with a technical setup to interact with the asvin.io infrastructure securely and efficiently. The asvin libraries act as an interface between an edge device and the asvin.io platform. The asvin.io SDK makes it easy to use APIs to interact with the asvin.io update infrastructure. The SDKs are developer friendly and yield decreased time to market for new IoT devices to connect with asvin.io infrastructure and take direct advantage of a ready to run update distribution network.
The key features of asvin SDK is that It is a generic solution. it is not limited to any specific hardware platform e.g. Arduino, ESP, STM, Arm Cortex-M3, M4 etc. and software protocols.

The asvin.io libraries installed on edge device provides following functionalities:

1. **Device Update Management**
   - securely register device on asvin.io platform

2. **Check for Firmware Updates**
   - regularly poll for updates in configurable time interval

3. **Download and Install Firmware**
   - download and store updated version of firmware
   - install the firmware based on traffic and availability of edge device

4. **Collect Health Statistics**
   - send timely health updates of edge device on asvin.io platform, e.g. Firmware Version in use
   - asvin.io platform generates insights from the data which can be used to extend life cycle of IoT device and for preventive maintenance.

Hyperledger Fabric Blockchain

The current blockchain network is built on Hyperledger Fabric. The fabric network enables to utilize the potential of Distributed Ledger Technology (DLT) and provides a framework to build distributed applications upon. The ledger stores all transactions executed on the asvin.io network: device register, firmware upload, device update, firmware update, user register etc. All these transactions are connected with hashes and stored in blocks which are again linked with a secured hash. This process provides security and immutability to the ledger.

The asvin.io platform is backed with a cluster of servers which runs fabric network to maintain millions of IoT devices. The asvin.io has built this cluster in such a way that it can be scaled up multifold to support increasing demand of edge devices. The fabric network is constructed of different pluggable modules which allow asvin.io infrastructure to be more flexible. This modular and pluggable approach enable asvin.io to develop tailored solutions for IoT customers. The cluster is formed using docker swarm. The fabric has following modules and services.

1. **Peer**
   It is an entity in the fabric network which receives request from applications, runs a chaincode, validates transactions, and maintains ledger.

2. **Orderer**
   This service orders all the transactions happening in the fabric net work and forward them to peers to be validated

3. **MSP**
   It takes care of providing digital identities to every member of the fabric network.

4. **CouchDB**
   Storage of current state ledger data.

The operating system level virtualization is achieved on blockchain server using docker. Each service in the fabric network runs in a separate docker container. These docker containers are hosted on multiple machines on the cluster. The communication among containers is achieved using docker swarm. The whole fabric network is developed, deploy and run using docker swarm technology.

Customer Platform

At asvin.io, we believe in a simple and user-friendly solutions. asvin.io provides an abstraction layer to hide the complexities and sophistication of the Distributed Ledger technology. This abstraction is facilitated using a cluster of servers backed with database server. To cope up with gigantic number of IoT devices a load balancer is installed. The load balancer streamlines the connection from customers and IoT edge devices to the server.

The customer Platform delivers following functions:

1. **Service Dashboard**
   asvin.io.io provides a portal for the IoT device Operators to directly in teract and control the update and patch status of their edge devices.
   The portal has functionalities to upload firmware, delete firmware, manage edge devices, check health statistics of edge devices etc.

2. **Upload Firmware**
   Uploads firmware provided by customer to IPFS Server

3. **Update Ledger**
   interacts with Hyperledger blockchain server to update firmware database.

4. **Update Version Control Server**
   it keeps version control server updated with information of latest firmware.

Version Controller

To manage billions of IoT edge devices asvin.io.io employs a distributed service cluster infrastructure. For edge devices this complexity is invisible, and they interact with the cluster as they would do with a single machine. The version controller server is consist of multiple nodes which has same copy of web server and hosts identical web services. Each node in the cluster is fully functional web server and can serve a request independently. Each node has different IP adress, but they are hidden to edge devices. An abstraction layer is used on top of the cluster to hide complexity. The abstraction makes use of round-robin DNS[2] technique for load balancing, fault tolerance and load distribution. The server accepts DNS requests and responds to them by forwarding it to a computing machine in the cluster. A machine from the cluster is chosen in round-robin fashion.

[2] https://en.wikipedia.org/wiki/Round-robin_DNS

The key features of the version controller server are following.

High-availability

The two layers architecture of the version controller server provides defense against single point failure. Even if a node in the cluster goes down because of hardware or software problem the asvin.io framework stays functional.

Scalability

It is easy to install a new node in the cluster to enhance performance of the version controller server. The cluster is highly scalable and provides stability.

Efficiency

The workload is distributed among multiple web server in the cluster that gives boost to the network performance, reduce collision and avoid congestion during high demand.

The Version controller provides resilience to the asvin.io network. The server performs following tasks.

1. **Response to Edge Devices**
   The edge devices poll the version controller to check for new update. The server re sponds to edge device with information of a valid firmware.

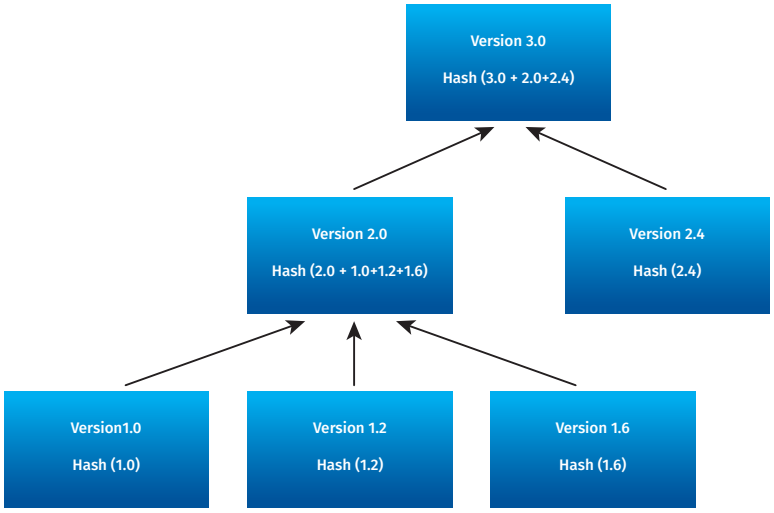
2. **Latest Firmware Version List**
   It maintains real time information of different versions of firmware available on data storage servers. The version controller has a list of available firmware on asvin.io plat form for all edge devices. It keeps the list updated by interacting with Customer plat form server.

InterPlanetary File System

asvin.io uses Interplanetary File System protocol to store firmware and patches. The IPFS is a content-addressable peer to peer method for storing and sharing hypermedia in a distributed file system. It uses a network infrastructure that enables asvin.io to store unalterable firmware data. As mentioned above it is content addressable protocol which solves problem of duplicate files across the network and remove redundancy.

When a firmware file is stored on the network a hash is generated based on content of the firmware and stored on blockchain network. Later on, the same hash is used by edge devices to pull the firmware from data storage. This forms a generalized Merkle directed acyclic graph(DAG). Each node of Merkle DAG is connected with a secured Hash. When a node is added in the DAG its hash is computed based on hash of its local content and hashes of its children's name instead of their content. Once it is created, it is impossible to alter a node in the network. IPFS has no single point of failure. asvin.io provides a distributed content delivery system. This facilitate an extra layer of security and prevent DDoS attacks.

asvin.io SDK which is run on edge devices enables the functionality to interact with data storage servers. Once an edge device gets information from the version controller regarding the newly available firmware, it uses this information to download firmware from the distributed CDN.



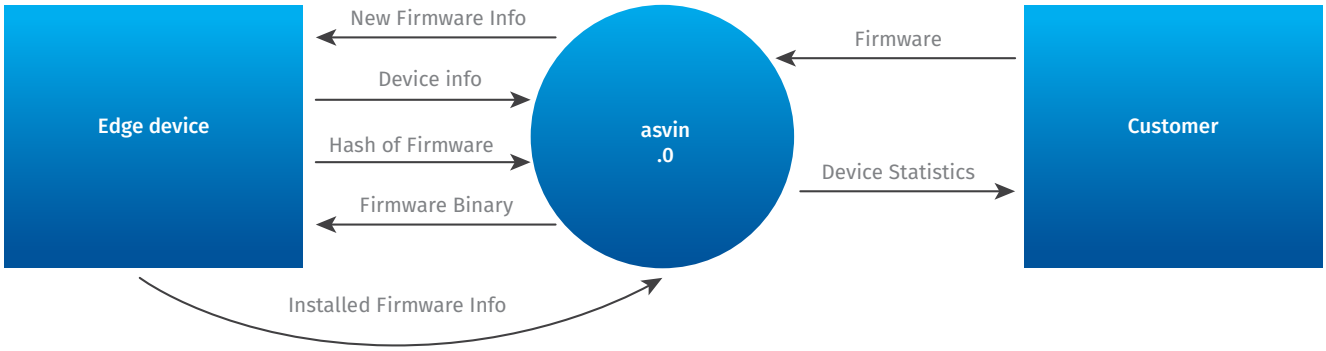
Data Flow Diagram

Data flow diagram represents data floating around in a system among multiple processes which transform the data. Here the DFD is a graphical overview of the asvin.io platform. The asvin.io platform encompasses distinct processes to handle requests from edge devices. The DFD captures a broad picture of what sorts of data is taken in and out by the data processes, how the information moves forward among processes, and where the data is accumulated. It helps to visualize the structural design aspect of the asvin.io computing model.

A DFD follows top to bottom approach to system design. It consists of many well-defined diagrams and each diagram has different level of information. The first model is called "Context Diagram" and the main process in it is named '.0'. The next diagram is called top level DFD and titled DFD level 0. It gets its name from main process. The processes defined in top level DFD are consecutively by natural numbers beginning with 1. When a process is exploded in a lower level DFD, the processes in such lower level DFD are consecutively numbered following the label of such parent process ending with a period or full-stop, for an instance 1.2, 1.2.3 etc.

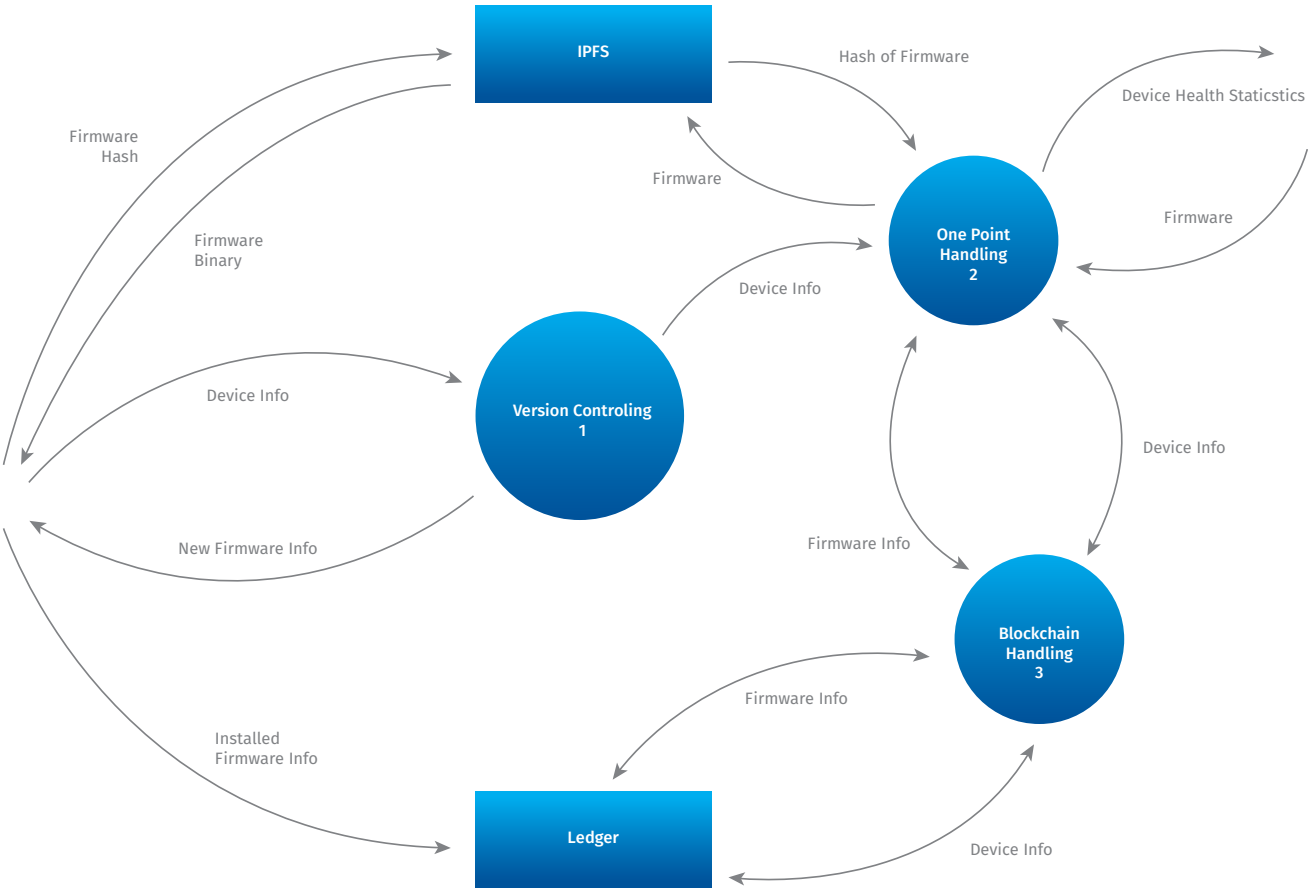
Context Diagram

It visualizes the whole system as a process and depicts eternal actors interacting with the main process, and the data that is being exchanged between the platform and its users. As the diagram shows asvin's customers upload firmware on the platform and edge devices are monitored, control and updated with the firmware available to be installed.



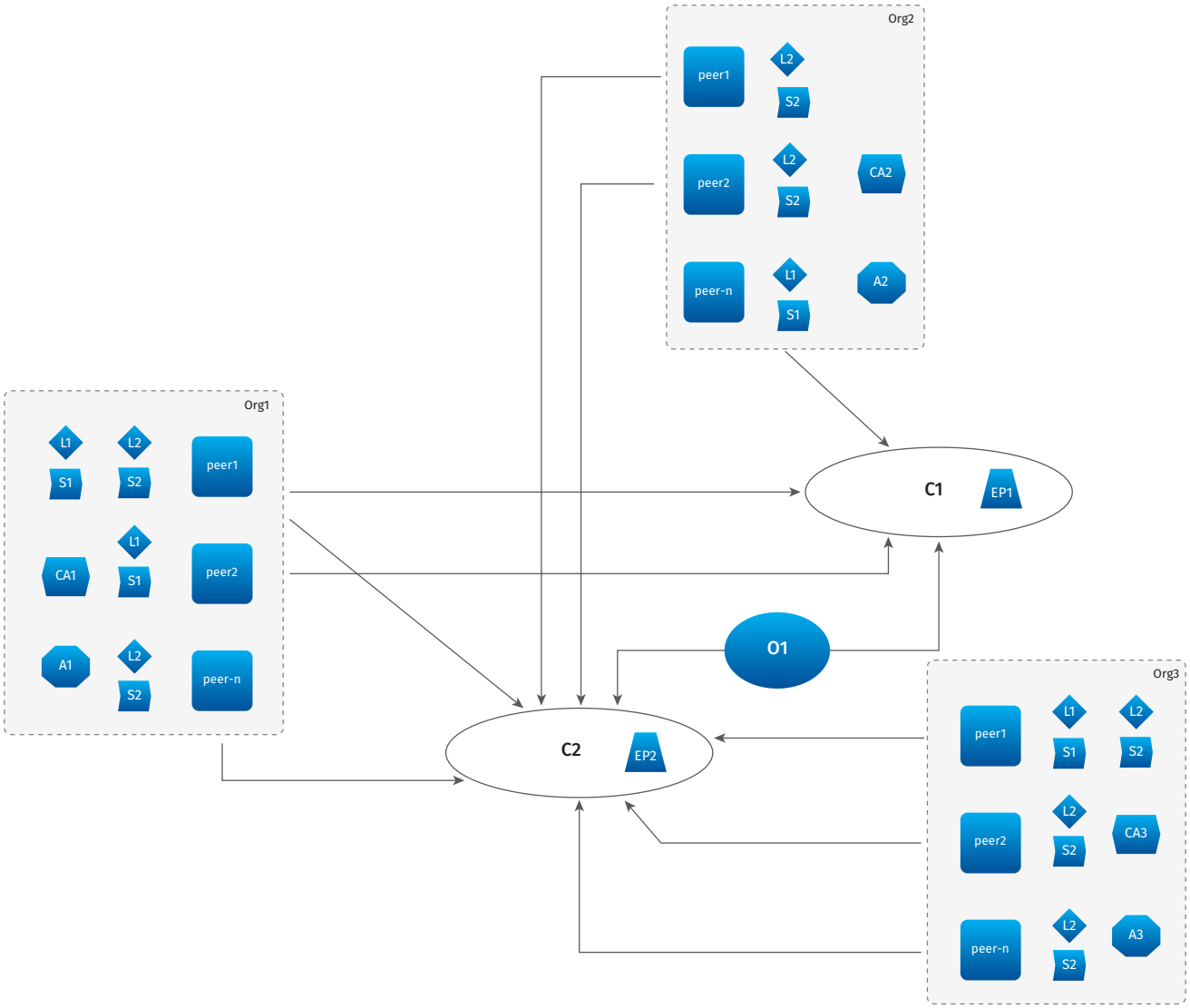
DFD0

It shows all the top level processes on asvin.io platform. Each process has its well-defined task.



Distributed Ledger Technology

To provide secured framework for firmware update here at asvin.io, we use Hyperledger fabric[3]. The Fabric provides permissioned blockchain network.



Fabric Components

Organization

Fabric gives flexibility of implementing blockchain technology using simple business logic. For an instance a trade framework is composed of multiple companies which transact a product/ service with each other. Similarly, these companies are implemented as Organization(org) in the asvin.io fabric network.

[3] https://hyperledger-fabric.readthedocs.io

Peer

Peers are the entities in an organization which maintain distributed ledger. The peers run chaincode (smart contract) in order to perform read/write operations on the ledger. Each peer in the network has a consistent copy of the ledger. The peers are owned by an organization. An organization can have multiple peers.

Channel

A channel in the network is a route which provides confidentiality and isolation. This concept set apart asvin.io fabric network from traditional cryptocurrencies blockchain like Etherium, where there is no concept of private transaction among multiple parties. In the asvin.io fabric network peers from multiple organizations are connected through channel. A separate ledger is maintained for each channel. A channel is governed by policies decided by organizations.

Orderer/Ordering Service

It is an administrative center for the fabric network. This encompasses channel configuration information. This configuration is comprised of policies for the channel and membership information for each member of the channel. The ordering service orders transactions happening on the network on a first-come-first-serve basis and pack them into blocks, which are then then sent to peers for validation based on endorsement policies. After validation ledger is updated and event is generated. Ordering service is a pivotal point in the fabric.

Membership Service

To provide security to the network all the nodes are given digital identities e.g. org, peer, orderer and policies are created for channels. It uses certificate authority (CAs) to provide certificates. For a node msp defines permissions and roles. Channel configuration defines administrative and participatory rights at the channel level.

Client Application

Client applications provide user interface to organizations(org) to interact with blockchain network and ledger. An application is owned by org. It is the place where transaction proposals are generated and sent to peers connected on the channel which execute chaincode and send response to applications against their copy of ledger. Client application interact with peers and ordering service on the channel.

Security Landscape

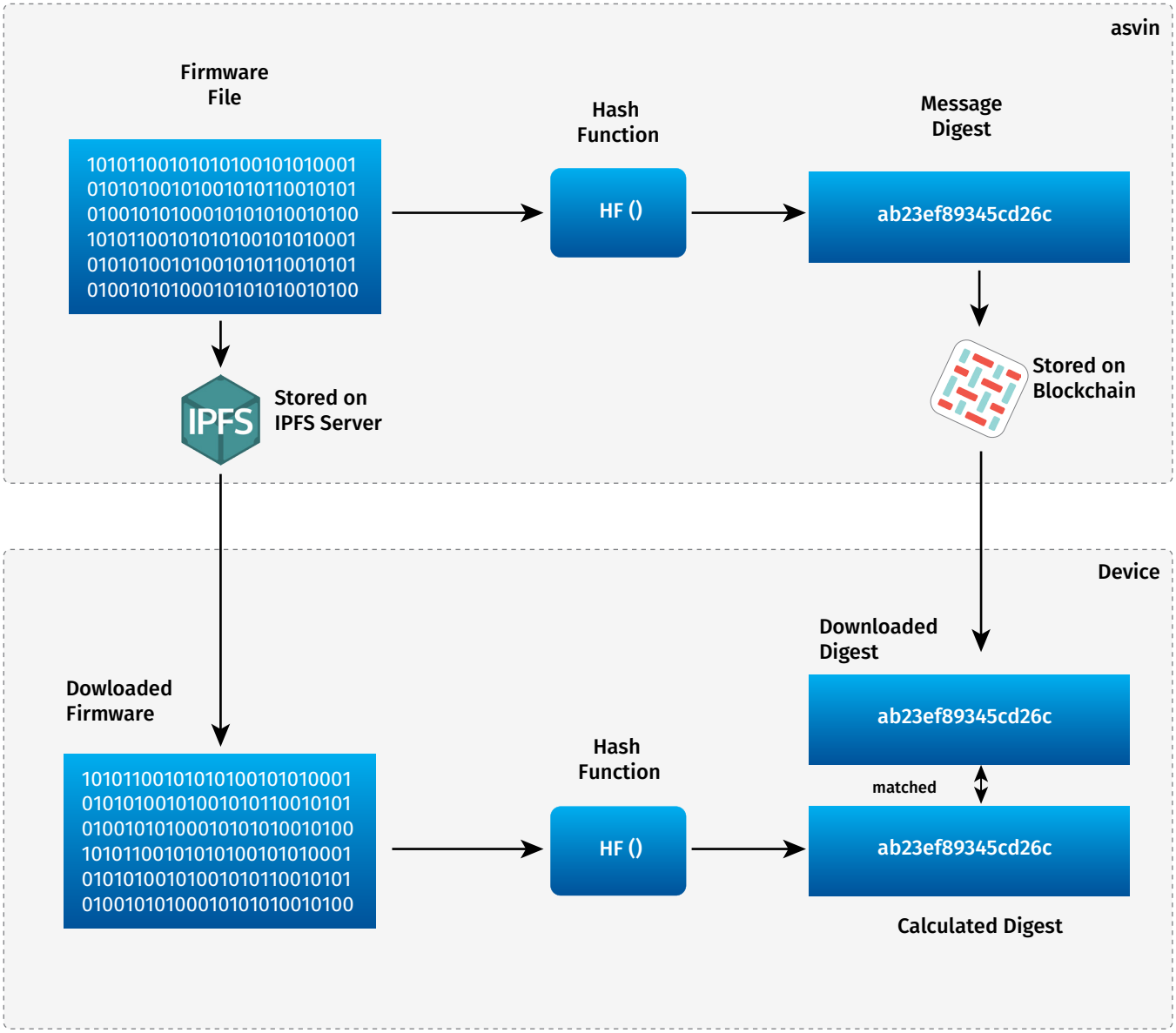
Security of a firmware update platform is the foremost objective. A compromised update mechanism can provide a route to infuse vulnerabilities to the connected IoT edge devices. Even a vulnerable IoT device is a substantially liability to its customer, manufacture and whole network. A single security breach in a network might leads to service unavailability and data & privacy breach to customers and enormous financial, corporate credibility loss to an organization. The asvin.io addresses broad spectrum of security threats present in the existing software update solutions. There are multiple characteristics of asvin.io which make it more secure, resilient, and threats proof.

DoS Filter

A denial-of-service(DoS) attack is a malicious attempt to disrupt and eventually shut down a service, server or network for its legitimate users by flooding the target with unintended traffic. Though the DoS attacks do not provide access to privileged resources or result in data or privacy breach, they can incur a significant amount of time and money on an organization to suppress the attack. In a distributed denial-of-service(DDoS) attack the flood of traffic is originated from numerous distinct sources.
The trends in the malicious attacks confirms that the DoS attacks continue to multifold their strength and frequency. The attacks are more sophisticated to detect and restrain using existing security models. The asvin.io incorporates a DoS filter which provide defense against all sorts of DDoS attacks. The asvin.io's DoS filter employ various techniques to foil all attempts of a perpetrator to disrupt the service. All traffic is passed through the DoS filter, which scan for multiple parameters e.g. illegitimate content & intent, ports and IP addresses of sources, number of requests generated etc. Once an abnormal activity is detected the DoS filter takes appropriate actions for instance limiting number of requests, blocking and blacklist the malicious IPs etc. to overcome the situation.

Cryptographic Hash

The data integrity of the firmware images is preserved on asvin.io platform throughout the entire lifecycle. A typical lifecycle of a firmware begins when a customer uploads a new version of firmware on asvin.io. It is stored on IPFS server and once rolled out it is downloaded by an IoT edge device. This means that a firmware file cannot be altered in an unauthorized or undetected manner on asvin.io platform during storage and transmission. asvin.io conserve and assure the accuracy and completeness of all firmware. Data integrity of the firmware is ensured using cryptographic hash function. The hash function is designed to be a one-way function, which means it cannot be used to reverse the operation. A message digest is computed for each firmware file housed on asvin's server using the hash function. The firmware file is stored on IPFS server and the respective message digest is stored on blockchain server.
An IoT edge device download a new version of a firmware using IPFS apis and the message digest from the Hyperledger blockchain network. The device computes message digest for the downloaded firmware and compare it with the downloaded one. If both the message digest matches, then it is ensured that the firmware file is not altered.


asvin
Firmware File
10101100101010100101010001
01010100101001010110010101
01001010100010101010010100
10101100101010100101010001
01010100101001010110010101
01001010100010101010010100
Hash Function
HF ()
Message Digest
ab23ef89345cd26c
IPFS
Stored on IPFS Server
Stored on Blockchain
Device
Downloaded Digest
ab23ef89345cd26c
Dowloaded Firmware
10101100101010100101010001
01010100101001010110010101
01001010100010101010010100
10101100101010100101010001
01010100101001010110010101
01001010100010101010010100
Hash Function
HF ()
matched
ab23ef89345cd26c
Calculated Digest

Digital Identity

asvin.io uses Hyperledger Fabric blockchain network to construct an extra layer of security. All the sensitive information for instance IoT edge device details, firmware characteristics, message digests, IPFS hashes, customer details etc. are kept on blockchain network and a distributed ledger is maintained. All interactions of devices and customers on the asvin.io platform is validated and logged on blockchain server.
The Hyperledger Fabric uses a component called Membership Service Providers (MSP). In nutshell, MSP is an abstraction layer on top of all cryptographic mechanisms and protocols. Behind the curtain, MSP uses cryptographic algorithms to issue and validate certificates. These certificate serves as digital identities for user authentication. All operations and events are governed using the digital identities.
MSP has a modular design and it is used to configure the parameters for desired security requirements. These parameters are worked out by RFC5280[4]. The Hyperledger Fabric uses X.509 certificate for digital identity. In cryptographic model, X.509[5] is a standard to define the format of public key certificates. A X.509 certificate accommodates information regarding the identity to which a certificate is issued and the identity of Certificate Authority which issued it. It is used in internet protocol TLS/SSL , which is the foundation of HTTPS.

Web Application Firewall

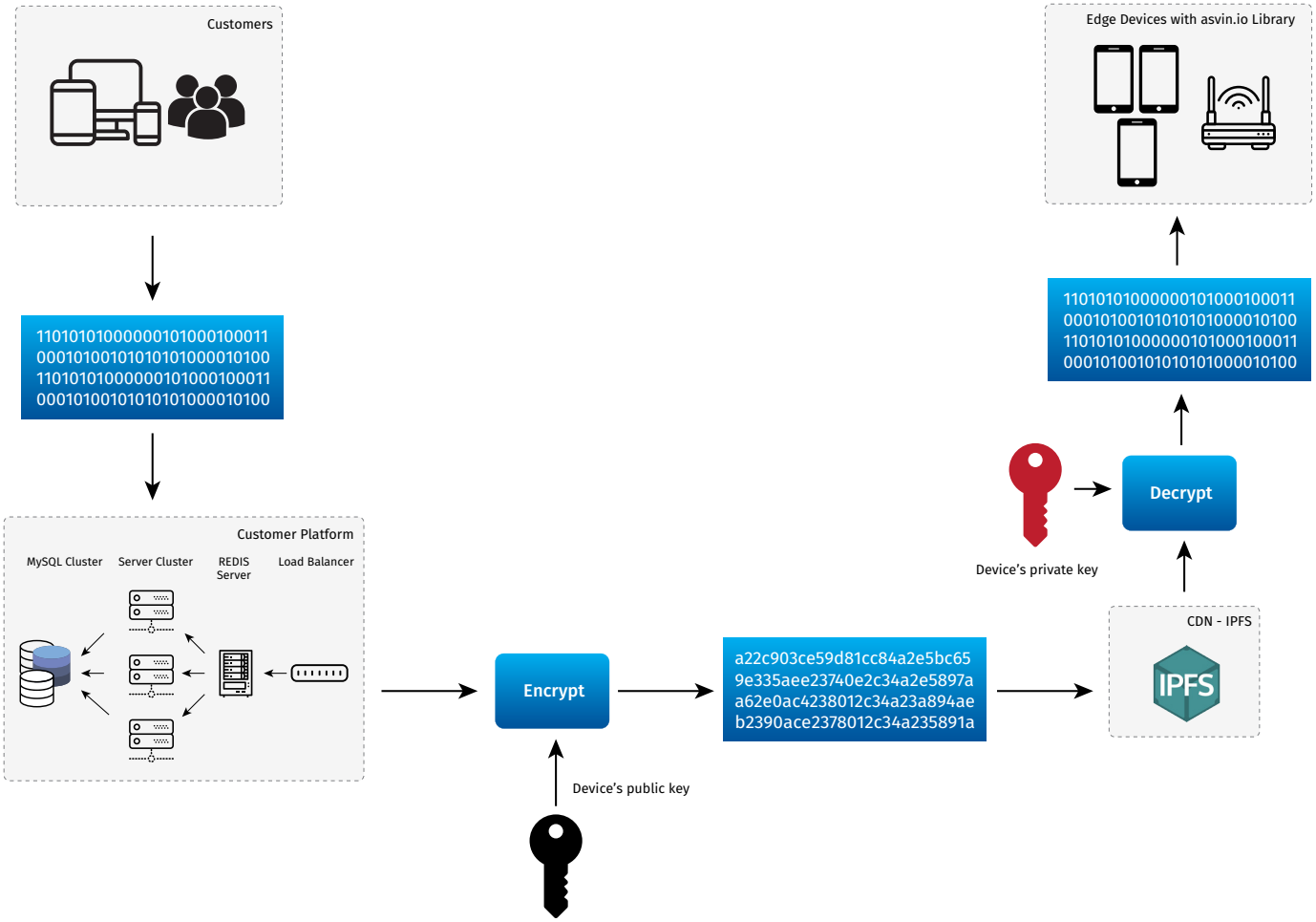
A Web Application Firewall (WAF) plays a crucial role as security defense by filtering, monitoring and blocking HTTP traffic in asvin.io. It is deployed in front of the asvin.io app running on edge devices. It acts a shield between the application and the internet. A WAF analyzes bi-directional HTTP traffic and detect any malicious activities.

A WAF provide defense against multiple security attacks such as SQL injection, Cross-site scripting (XSS), file inclusion etc. It is configured through a set of rules, also known as policies. A specific policy is constructed for a security attack. The multiple policies are written to provide a tailored solution.

Firmware Encryption

Firmware and patches for IoT devices are the central part of the update management platform. It is not desirable to store the firmware on cloud storage without encryption. In a case of security breach on storage server, firmware can be stolen or compromised and attackers might get access to an intellectual property (IP) of an organization using the firmware. When devices are registered on asvin.io platform a pair of public and private key is generated for a group of devices. A group of devices which share a same firmware are given identical keys. A firmware is encrypted with public key of respective group of devices before pushing it to the IPFS server and decrypted by private key before installing on a device. This ensure that content of firmware is always protected.

[4] https://www.ietf.org/rfc/rfc5280.txt
[5] https://www.itu.int/rec/T-REC-X.509/en


Customers
11010101000000101000100011
00010100101010101000010100
11010101000000101000100011
00010100101010101000010100
Customer Platform
MySQL Cluster
Server Cluster
REDIS Server
Load Balancer
Encrypt
Device's public key
a22c903ce59d81cc84a2e5bc65
9e335aee23740e2c34a2e5897a
a62e0ac4238012c34a23a894ae
b2390ace2378012c34a235891a
CDN - IPFS
IPFS
Device's private key
Decrypt
Edge Devices with asvin.io Library
11010101000000101000100011
00010100101010101000010100
11010101000000101000100011
00010100101010101000010100

Credits

Publisher
Asvin GmbH
Schulze-Delitzsch-Str. 16
70565 Stuttgart
Germany

www.asvin.io
contact@asvin.io

Authors
Rohit Bohara
Mirko Ross
Sven Rahlfs

Version
Version of this Whitepaper 0.9.2