

Platform information system

Rautatieaseman informaatiotaulu järjestelmä

Samuel Laisaar, Leevi Laaksonen, Aleksandr Liski, Leo
Lehtiö

Platform information system

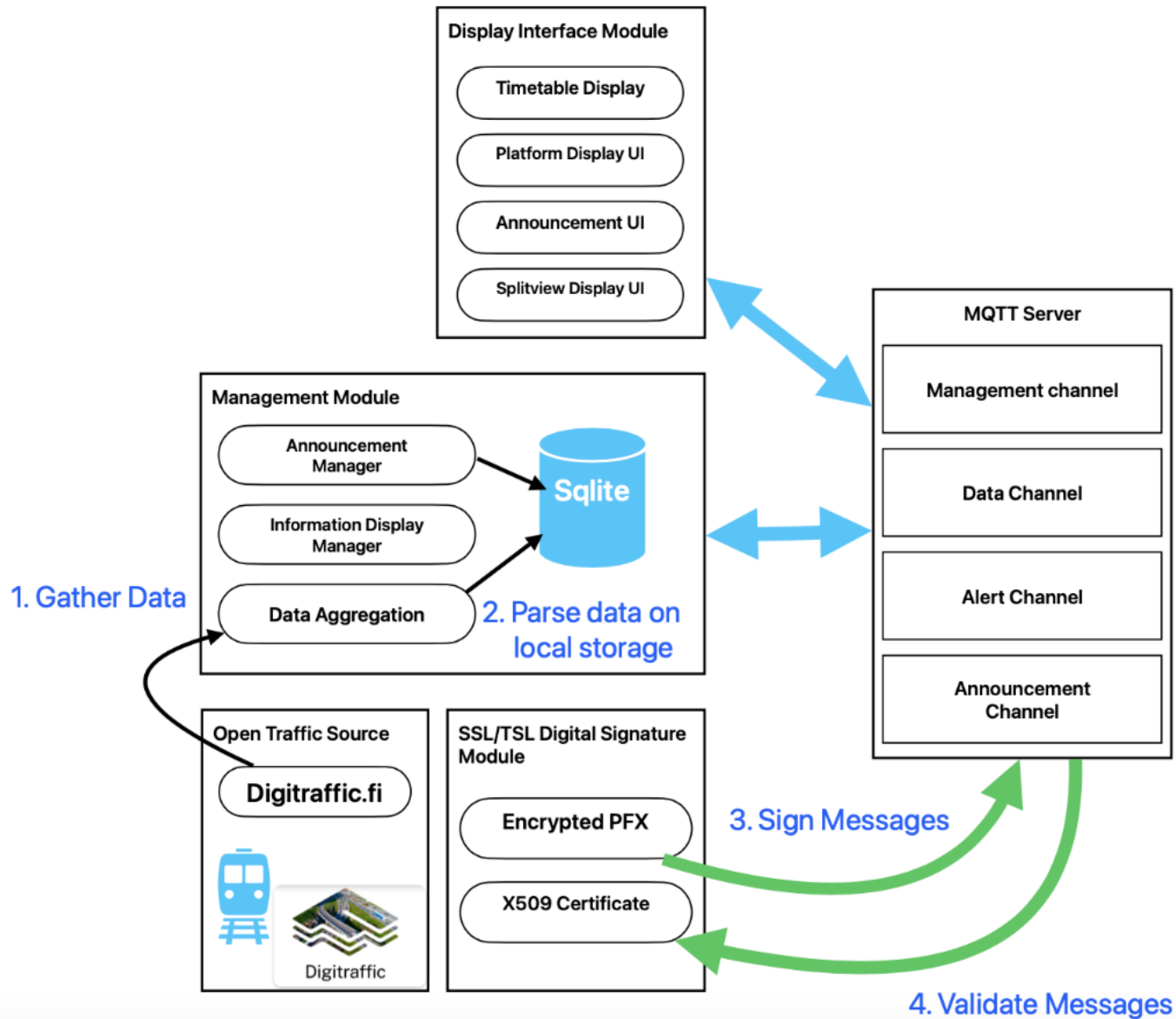
Rautatieaseman informaatiotaulu järjestelmä

- Mitä?
 - Junatietojen esittäminen & hallitseminen näytöillä
 - Digitraffic.fi:n avoin junadata
- Miksi?
 - Selkeä junatietojen esittäminen
 - Validoitu ja aito informaatio



Pasila asema departing commuter trains				18:30:26
Time	Notice	Train	Platform	Destination
18:31		E	8	Kauklahti asema
18:31	→ 18:35	K	1	Helsinki asema
18:32		P	11	Lentoasema
18:33		I	10	Helsinki asema
18:35		K	2	Kerava asema
18:36	→ 18:38	P	1	Helsinki asema
18:37		A	11	Leppävaara
18:38		A	10	Helsinki asema
18:40		I	2	Lentoasema
18:40		Z	4	Lahti

Järjestelmän suunnittelu ja arkkitehtuuri



- 4 päämoduulia
 - Display - Näytöt
 - Management - Hallinnointi & aggregointi
 - MQTT-viestintäprotokolla
 - SSL/TSL-varmenne datan validointiin
- Junatiedot - Digitraffic.fi avoin rajapinta

MQTT-viestintäprotokolla laitteiden väliseen kommunikointiin



- Topicit:
 - Junadata:
 - station/{station_code}/{platform_id}/{transit}/{transport}
 - Esimerkki: station/PSL/4/Departure/Commuter
 - Laitteiden väliseen viestintään
 - Management/{display_id}/update
 - Aseman ilmoituksia varten:
 - Announcement/{notifyType}/{station_code}/{platform_id}
 - Esim. Myöhästymiset, peruutukset yms.

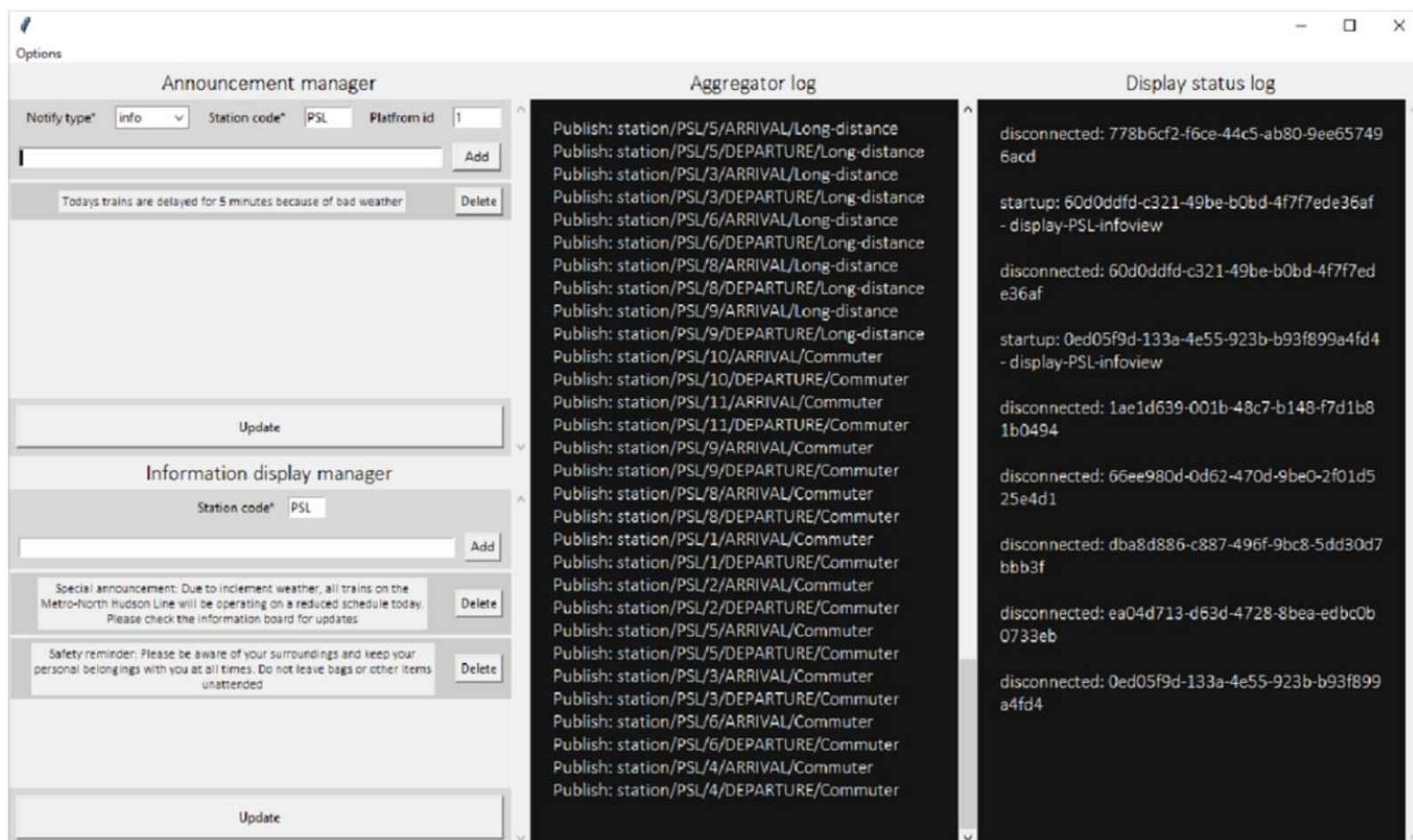
Aggregator

- Toimii datan julkaisijana, joka välittyy näytöille.
 - Julkaisee:
 - Tietoa junien aikatauluista
 - Ilmoituksia joita Manager ohjelmalla voi lisätä.
- Suorittaminen tapahtuu yhdestä päälaitteesta (ManagementNode)
- Käynnistetään komennolla:
 - `pis-aggregator <station_code> <station_code2> ...`
- Aggregator voidaan käynnistää jakamaan dataa usealle asemalle kerrallaan.



Manager

- Aggregatorin rinnalla toimiva ohjelma.
- Toimintoja
 - Näyttöjen ilmoitusten hallitseminen
 - Aggregatorin lähettämien aiheväylien (topic) seuranta.
 - Näyttöjen tilojen seuranta.
- Käynnistys: pis-dashboard



Näytöt

- 4 erilaista näyttötyyppiä
 - **Taulukkonäkymä (tableview)**
 - Seuraavan junan näkymä (platformview)
 - Kahden laiturin näkymä (splitview)
 - Informaatio näkymä (infoview)

- Käynnistetään komennolla:

- python3 display_client.py
 - view <display_view>
 - s <station_short_code>
 - p <platform>
 - left <platform>
 - right <platform>
 - transit <transit>
 - transport <transport>

Tableview esimerkki:

Pasila asema departing trains				22:35:40
Time	Notice	Train	Platform	Destination
22:36		P	1	Helsinki asema
22:37		L	11	Kirkkonummi
22:38		A	10	Helsinki asema
22:40		I	2	Lentoasema
22:42		P	11	Lentoasema
22:42		K	1	Helsinki asema
22:43		U	9	Helsinki asema
22:44		R	5	Helsinki asema
22:45		R	3	Tampere asema
22:51		IC972	9	Helsinki asema
Juna- ja lentoliikenteessä yhä häiriöitä huonon sään vuoksi				

Näyttötyyppi

Aseman koodi

Aseman laituri

Aseman laituri (Kahden laiturin näkymä)

Aseman laituri (Kahden laiturin näkymä)

Kulkusuunta (departures, arrivals)

Kuljetus (commuter, long_distance)

Näytöt

- 4 erilaista näyttötyyppiä
 - Taulukkonäkymä (tableview)
 - **Seuraavan junan näkymä (platformview)**
 - Kahden laiturin näkymä (splitview)
 - Informaatio näkymä (infoview)

- Käynnistetään komennolla:

- `python3 display_client.py`
 - view <display_view>
 - s <station_short_code>
 - p <platform>
 - left <platform>
 - right <platform>
 - transit <transit>
 - transport <transport>

Näyttötyyppi

Aseman koodi

Aseman laituri

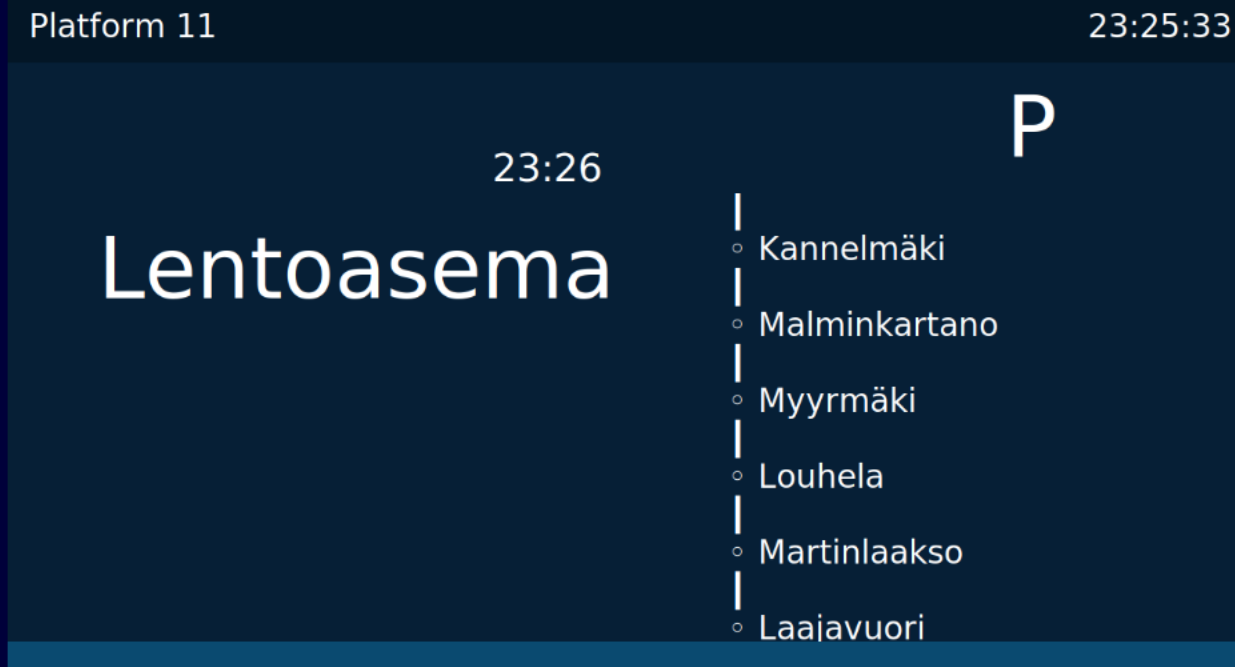
Aseman laituri (Kahden laiturin näkymä)

Aseman laituri (Kahden laiturin näkymä)

Kulkusuunta (departures, arrivals)

Kuljetus (commuter, long_distance)

Platformview esimerkki:



Näytöt

- 4 erilaista näyttötyyppiä
 - Taulukkonäkymä (tableview)
 - Seuraavan junan näkymä (platformview)
 - **Kahden laiturin näkymä (splitview)**
 - Informaatio näkymä (infoview)

- Käynnistetään komennolla:

- python3 display_client.py

-view <display_view>
-s <station_short_code>
-p <platform>
-left <platform>
-right <platform>
-transit <transit>
-transport <transport>

Näyttötyyppi

Aseman koodi

Aseman laituri

Aseman laituri (Kahden laiturin näkymä)

Aseman laituri (Kahden laiturin näkymä)

Kulkusuunta (departures, arrivals)

Kuljetus (commuter, long_distance)

Splitview esimerkki:

Pasila asema trains		22:36:42
22:40		22:45
1		R
2		3
Lentoasema		Tampere asema

Näytöt

- 4 erilaista näyttötyyppiä
 - Taulukkonäkymä (tableview)
 - Seuraavan junan näkymä (platformview)
 - Kahden laiturin näkymä (splitview)
 - **Informaatio näkymä (infoview)**

- Käynnistetään komennolla:

- python3 display_client.py
 - view <display_view>
 - s <station_short_code>
 - p <platform>
 - left <platform>
 - right <platform>
 - transit <transit>
 - transport <transport>

Infoview esimerkki:

Info

22:36:50

◦ HSL-sovelluksesta löydät helposti liput ja reitit. Osta kerta-, kausi- tai vuorokausilippu missä ja milloin vain ja maksa kätevästi maksukortillasi.

◦ Junalippujen hintoihin alkuvuodesta viety joukkoliikenteen alv-alennus on voimassa vielä huhtikuun loppuun asti! Voit siis ostaa junalippuja verovapaasti niin pitkälle tulevaisuuteen, kuin niitä on myynnissä, kunhan teet sen ennen vappua!

◦ Tiesithän, että voit muuttaa junalippuasi helposti VR Matkalla -sovelluksessa tai vr.fi:ssä, jos ostit sen kirjautuneena? Voit myös lisätä kirjautumatta ostetun lipun VR-tilillesi käyttämällä lipun tilaus- ja viitenumeroa.

Näyttötyyppi

Aseman koodi

Aseman laituri

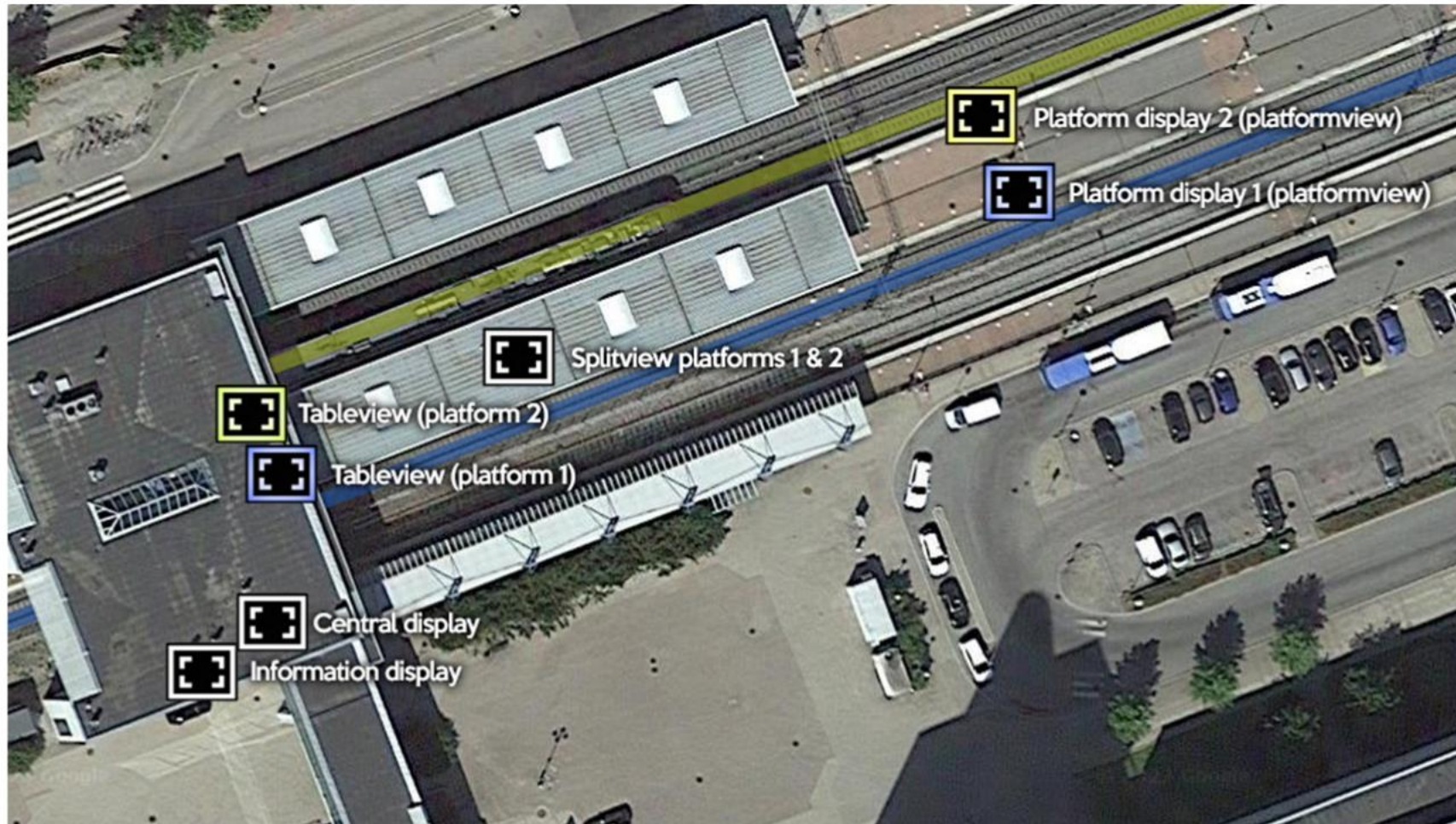
Aseman laituri (Kahden laiturin näkymä)

Aseman laituri (Kahden laiturin näkymä)

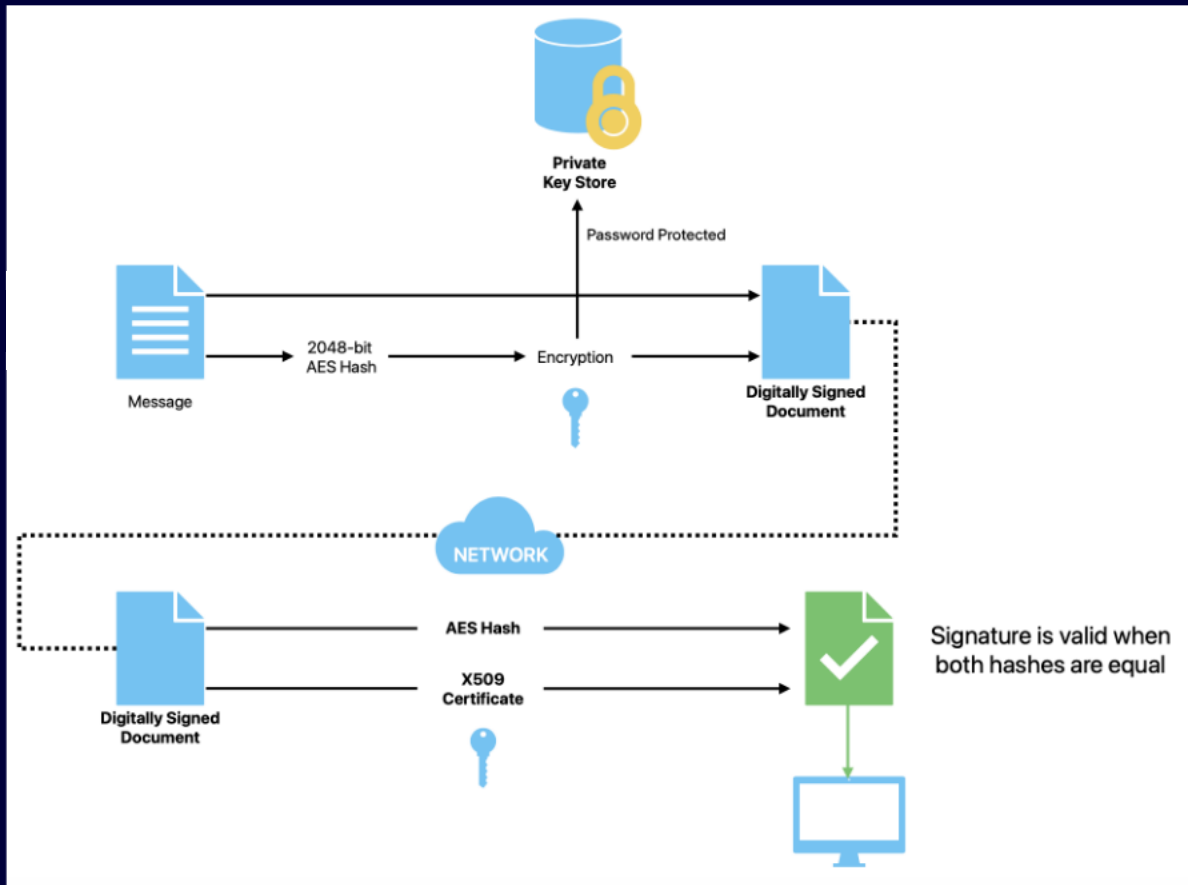
Kulkusuunta (departures, arrivals)

Kuljetus (commuter, long_distance)

Demo: Espoon rautatieasema



Validointi



- Alunperin oli pyydetty TPM2:n turvapiirin käytön viestinnän turvaamiseksi.
 - Yhteensopimaton kehitysympäristö
- Ratkaisuksi OpenSSL Digital Signature
 - Jäljittelee TPM:n käytöstä sopivasti.
 - Avain suojataan turvallisessa PKCS12 säiliössä
- Aggregaattorin lähettämiin junaliikennetietoihin lisätään digitaalinen allekirjoitus, jonka näytöt todentavat.



Haasteita ja mitä opimme?

Haasteet

Välillä koimme Digitrafficin avoimen datan käytön hankalaksi

- Datan formatointia tulee tehdä paljon
- Digitrafficin datan ymmärtäminen
 - Junien aikataulujen järjestäminen
 - Tiettyjen junien määränpäätt (esim. lentoasema)
 - Vanhojen junien poisto

Kehitysympäristön ja asiakkaan ympäristön eroavaisuuden koettui hankalaksi demoamis vaiheessa.

Uudet teknologiat aiheuttivat paljon opettelemista ennen toteuttamista.

Opitut taidot

- MQTT Protokollan hyödyntäminen IoT - laitteiden väliseen kommunikaatioon.
- Järjestelmän kokonaiskuvan suunnittelu.
- Perustellut tekniset ratkaisut. (esim. topics, datan formatointi)
- Pythonin virtuaalinen ympäristö ja Linux ympäristö tullut tutuiksi
- OpenSSL validointi
- "Suunniteltu ennemmin kuin koodattu"

Lopuksi

Dokumentaatio

Kaikki dokumentaatio löytyy projektin GitHub reposta doc kansiota:

<https://github.com/InnovationProject4/platform-info-system/tree/main/doc>

Sieltä löytää ohjeet, joiden perusteella järjestelmän jatkokehittäminen on mahdollista.



Jatkokehitysideoita

Näyttöihin voidaan lisätä:

- ★ Eri antureita, joiden dataa lähetetään aggregatorille.
 - ★ Kamera
 - ★ Enemmän telemetriaa, jonka perusteella aggregator saa enemmän tietoa näytöistä.
- Näytöt voivat toistaa ääntä. Kuten ilmoitus saapuvista ja lähtevistä junista.
- Mahdollisuus laajentaa myös järjestelmää toimimaan tie ja meri liikenteen kanssa.

Platform information system

Kysymyksiä?