Sign Language Translation Mobile Application and Open Communications Framework

Deliverable 2.3: First release of the SignON Open Cloud platform

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 101017255

Project Information
Project Number: 101017255
Project Title: SignON: Sign Language Translation Mobile Application and Open Communications Framework
Funding Scheme: H2020 ICT-57-2020
Project Start Date: January 1st 2021

Deliverable Information
Title: First release of the SignON Open Cloud platform
Work Package: WP 2 - SignON Service, Framework and Mobile application
Lead beneficiary: Dutch Language Institute (INT)
Due Date: 31/01/2022
Revision Number: V1.0
Authors: Marco van der Laan, Marcello Paolo Scipioni, John J O'Flaherty, Vincent Vandeghinste
Dissemination Level: Public
Deliverable Type: Demonstrator

Overview: The purpose of this document is to provide a technical description of the SignON Open Cloud platform.

Revision History

Version #	Implemented by	Revision Date	Description of changes
	Marco van der Laan	01-11-2021	First version, mostly hardware and virtualisation
	Marcello Paolo Scipioni	16/12/2021	Added section on Architecture of the Cloud Platform
	John J O'Flaherty	20/12/2021	Added subsection on the engineering test App.
	Marco van der Laan	29-12-2021	Added notes on use of GPU and some abbreviations and term explanations
v1.0	Vincent Vandeghinste	10/01/2022	Final read through and editing. Addition of conclusions and abbreviation list.

The SignON project has received funding from the European Union's Horizon 2020 Programme under Grant Agreement No. 101017255. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the SignON project or the European Commission. The European Commission is not liable for any use that may be made of the information contained therein.

The Members of the SignON Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the SignON Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Approval Procedure

Version #	Deliverable Name	Approved by	Institution	Approval Date
V1.0	D2.3	Aoife Brady	DCU	24/02/2022
V1.0	D2.3	Marcello Paolo Scipioni Marco Giovanelli	FINCONS	17/01/2022
V1.0	D2.3	Vincent Vandeghinste	INT	10/01/2022
V1.0	D2.3	Gorka Labaka	UPV/EHU	21/01/2022
V1.0	D2.3	John J O'Flaherty Ed Keane	MAC	11/01/2022 13/01/2022
V1.0	D2.3	Euan McGill	UPF	17/01/2022
V1.0	D2.3	Irene Murtagh	TU Dublin	21/01/2022
V1.0	D2.3	Mathieu De Coster	UGent	21/01/2022
V1.0	D2.3	Jorn Rijckaert	VGTC	xx/xx/202x
V1.0	D2.3	Ellen Rushe	NUID UCD	17/01/2022
V1.0	D2.3	Henk van den Heuvel, Louis ten Bosch	RU	21/01/2022
V1.0	D2.3	Catia Cucchiarini	TaalUnie (NTU)	21/01/2022
V1.0	D2.3	Myriam Vermeerbergen	KU Leuven	23/01/2022

V1.0	D2.3	Davy Van Landuyt	EUD	17/01/2022
V1.0	D2.3	Mirella De Sisto Dimitar Shterionov	TiU	18/01/2022 24/01/2022

Acronyms

The following table provides definitions for acronyms and terms relevant to this document.

Acronym	Definition
API	Application Programming Interface
ASR	Automated Speech Recognition
FTP	File Transfer Protocol
GB	Gigabyte
GPU	Graphical Processor Unit
HDD	Hard-Disk Drives
ISCSI	Internet Small Computer System Interface
ITIL	Information Technology Infrastructure Library
NFS	Network File System
NLP	Natural Language Processing
RAID	Redundant Array of Independent Discs
REST	Representational state transfer
SFTP	Secure File Transfer Protocol

SLR	Sign Language Recognition
SSD	Solid state drives
TB	Terabyte
VM	Virtual Machine
VPN	Virtual Private Network
WP	Work Package
Term	**Definition**
CUDA	CUDA is a software layer that gives direct access to the GPU's virtual instruction set and parallel computational elements, for the execution of compute kernels

Table of Contents

1. Introduction

This document describes the process of the design and implementation of a shared platform on which to host developing or developed parts of the SignON software and data. The platform consists of two separate entities: the repository with reference data and training data, and the hosting platform with processing space.

2. Repository

In March 2021 a storage system of 64TB with data was earmarked for SignON by INT in order to create a private repository in which SL data can be stored for project-internal use. The system was at the time not yet available, and an alternative was created using readily available dataspace and a frontend using secure FTP as the communication protocol. The server was prepared for external connections to the hosting system by implementing NFS server capabilities.

When the new hardware became available the server was migrated from an FTP based system to an SFTP based system which features encrypted data transport. SFTP login information has been passed on to all consortium partners that work with the datasets.

3. Hosting

The hosting platform is also located at INT and will host the central services of the final application. A detailed description of the architecture of the SignON app can be found in public deliverable D.2.2 SignON Services Framework Architecture, which also describes which parts will be hosted on a central server, such as the Orchestrator, the Message-Broker and several of the language specific analysis and generation components.

3.1. Requirement definition

In a meeting facilitated by FINCONS, for the purpose of discussing the development approach for D2.1 (SignON development Repository), a consensus was reached to use Docker as the main development container. Following this meeting a questionnaire was sent to the main developers involved in the project (WP3, WP4 and WP5) asking for the dimensions of the hardware they expected to need for their deliverables. The results of this questionnaire made it clear that a system heavy with computing power and storage was required. Memory was shown to be less critical.

3.2. Hardware choices

Several hardware providers were contacted with a request for a heavy processing machine with GPU and fast storage. After comparing the quotes, a suggestion was shared with the SignON consortium. This detailed a machine with 2x16 Intel Xeon 6346 cores, 2xT4 NVIDIA GPU, 256GB memory and a mixed storage with both SSD and HDD. The goldtype Xeon processors were chosen to facilitate workloads with a lot of contact switching, as experienced with virtualisation techniques like Docker or vmware-ESX type virtualisation.

Storage for the hosting platform will be built as a 15TB storage array with built-in redundancy against hardware error (RAID). RAID storage is known to be relatively slow, so for better performance 4TB fast storage is provided by SSD. Our first intention to use a tiered storage system has been under discussion because a tiered system does not offer advantages in processing large amounts of unique data.

Initially two smaller GPUs were suggested, assuming that parallel processes over two cores would be better than a single threaded power. After feedback, the raw power of the A40 GPU with 100% more CUDA-cores than two T4 GPUs was preferred by the stakeholders. This leaves room for expansion should more power ever be needed.

The delivery time was extended due to worldwide shortages in digital technology and COVID-related delays in the supply chain in late 2021.

3.3. Summary of hardware

Repository:

- HP Storageworks SAN platform
- 64TB storage in RAID 6 with extra hot standby
- Hardened Microsoft operating system with native ssh daemon
- NFS subsystem for sharing data
- 2x 1 GB Network Interface Card

Hosting platform

- 32 core Intel Xeon

- A40 GPU
- 256GB DDR4 RAM
- 4TB SSD Storage
- 15 TB Discs Storage, RAID5
- 4x 1GB Network Interface card

3.4. Operating system

The hardware will be taken up into the INT esx-based cloud. In the development phase, contributors will have their own VM with Docker stack. The repository will be available to each individual VM via an NFS or ISCSI connection. In later stages of development the separate VMs can be consolidated into a single Docker host if the different contributors consider this to be beneficial.

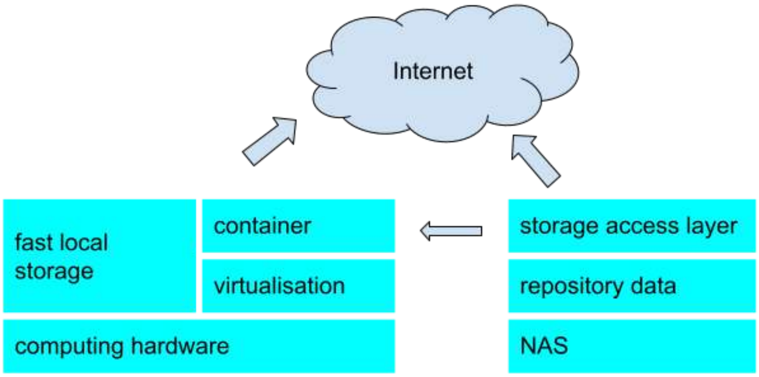


Figure 1. First design of the SignON infrastructure

3.5. Developer access

To facilitate developer access, a VPN access point was created. Using open source software a secure point-to-point connection can be established over the internet, routing traffic from the developers workstation to the management network used to access the VM.

The VPN software used is OpenVPN[1], which has been used successfully for the work-at-home environment of the INT. A central access point to a management network was created instead of a VPN on every individual container, as this is easier to manage.

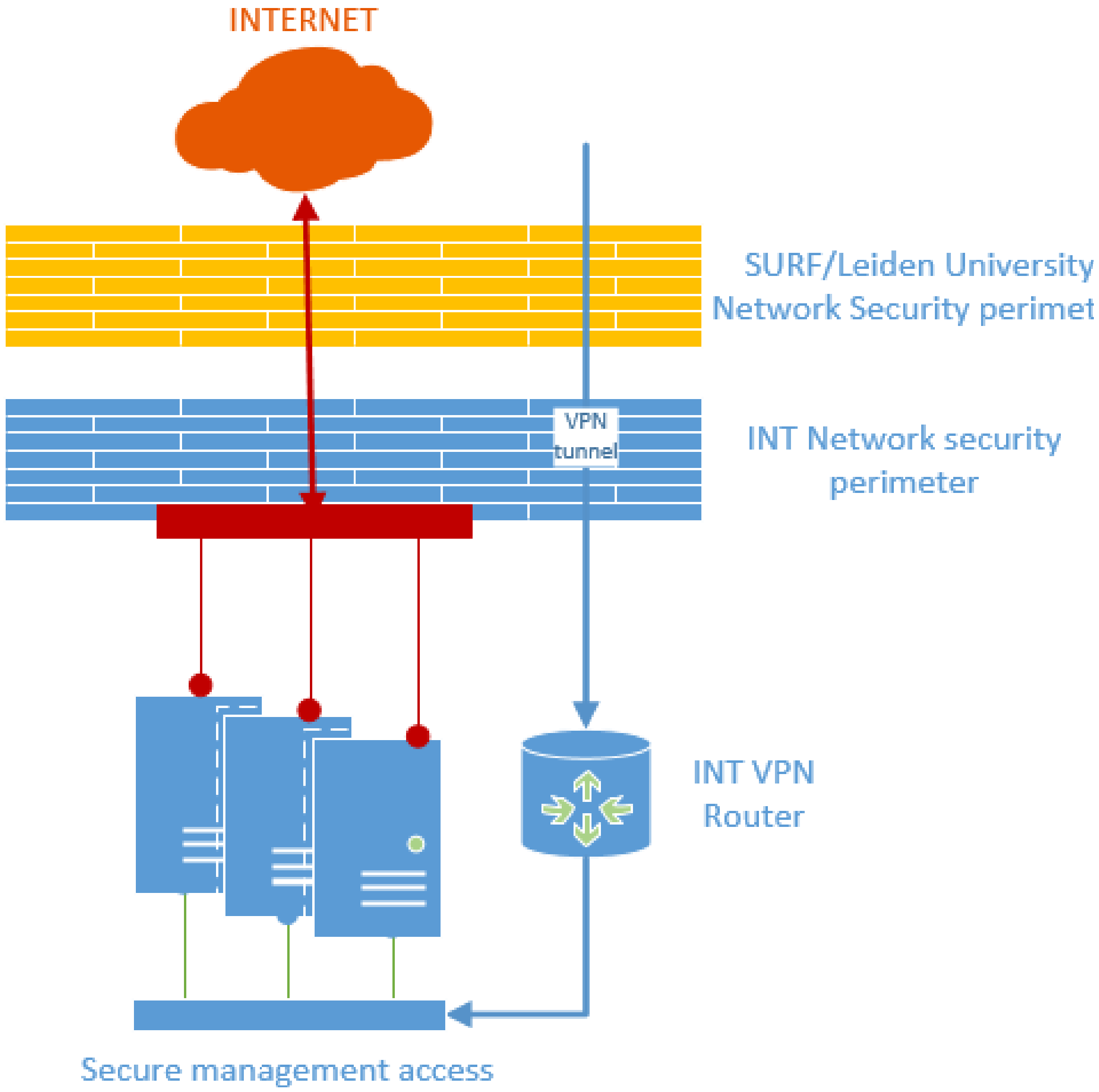

INTERNET
SURF/Leiden University Network Security perimet
VPN tunnel
INT Network security perimeter
INT VPN Router
Secure management access

Figure 2. VPN access point in the SignON infrastructure

[1] https://openvpn.net/

4. Security

Network security on the exposed systems is established by using multiple layers of network filtering using firewalls, and exposing as little as possible to the internet to minimize the attack surface. The platforms' operating systems are secured according to best practices, as recommended by Leiden University and SURF, the Dutch academic ICT cooperation organization. Brute force attacks are mitigated against by using the fail2ban[2] system that blocks network addresses after too many failed logins.

Security patches are evaluated on release to determine severity. Important patches are implemented as fast as possible, regular patches are implemented once a month. Regular external scans are done by security teams from Leiden University and SURF, to expose vulnerabilities and check compliance with the security policy of these organisations.

Physical access to the hardware is restricted to INT support personnel in a secured datacenter. System backups are made daily. Emergency recovery backups are kept for 2 weeks, long term data backups are kept for 7 years in a daily/monthly/yearly classic tiered backup schedule.

INT provides a team of system administrators to support SignON tenants. INT uses the ITIL process library[3] and uses email as their primary means of communication. Service level is best effort and during office hours.

[2] https://www.fail2ban.org/wiki/index.php/Main_Page
[3] https://www.servicenow.com/lpebk/itil4-guide.html

5. Architecture of the Cloud Platform

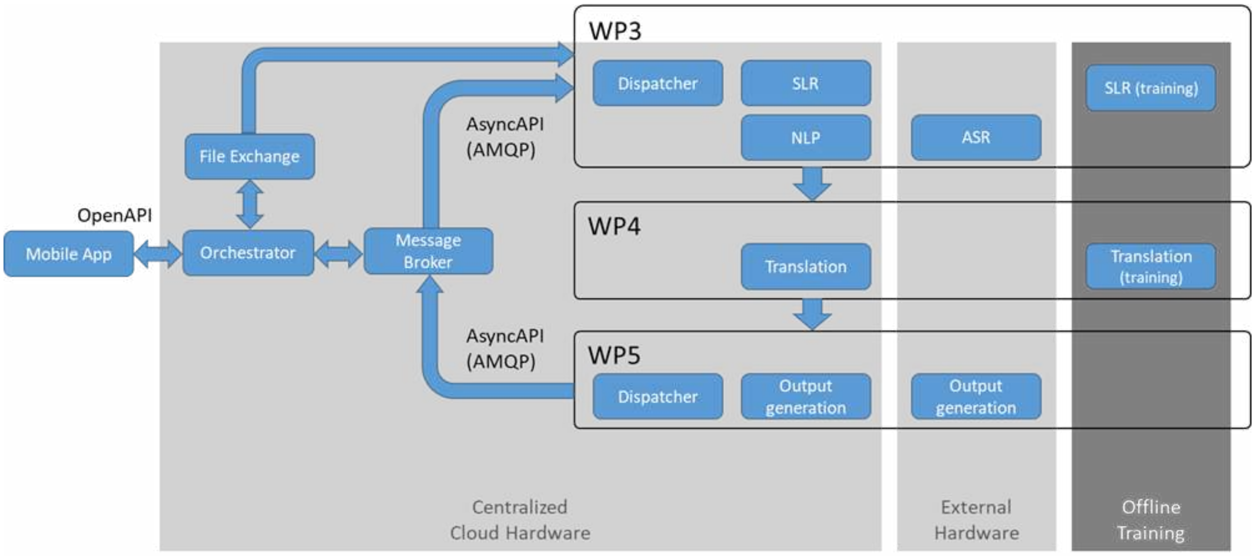


Figure 3. SignON Framework Architecture

Figure 3 represents the architecture of the cloud platform in terms of Docker containers and their functional role within the platform. Containers will have different roles: on the centralized cloud hardware containers will be hosted to perform online translation tasks, while some other Docker containers will run on external hardware; training activities needed to set up models employed at inference time are instead performed offline on dedicated containers.

The SignON framework is invoked from the SignON mobile App; the App connects to the framework via the SignON API. The source message (including relevant metadata) from the Mobile App is then processed by an Orchestrator, which queues it towards the processing pipeline with the help of a message broker, which decouples the communication with the pipeline. A Dispatcher on the WP3 side is subscribed to the appropriate queue and receives the message, invoking the relevant component depending on the type of input message received: SLR will be invoked if the source message is in Sign Language format, ASR if the source message is Audio format, and NLU if the source message is in Textual format (and after ASR, on the recognized text). After the required processing is complete, the message is passed to the next stage of the pipeline for the translation phase (WP4), along with its metadata. Finally, once the translation tasks have been completed, the message and metadata are sent to WP5, which produces the output message in the requested format (text, audio or sign language avatar). The output message is then delivered to the Orchestrator thanks to a new queue, on which a Dispatcher on the WP5 side writes the result of the processing on the pipeline side.

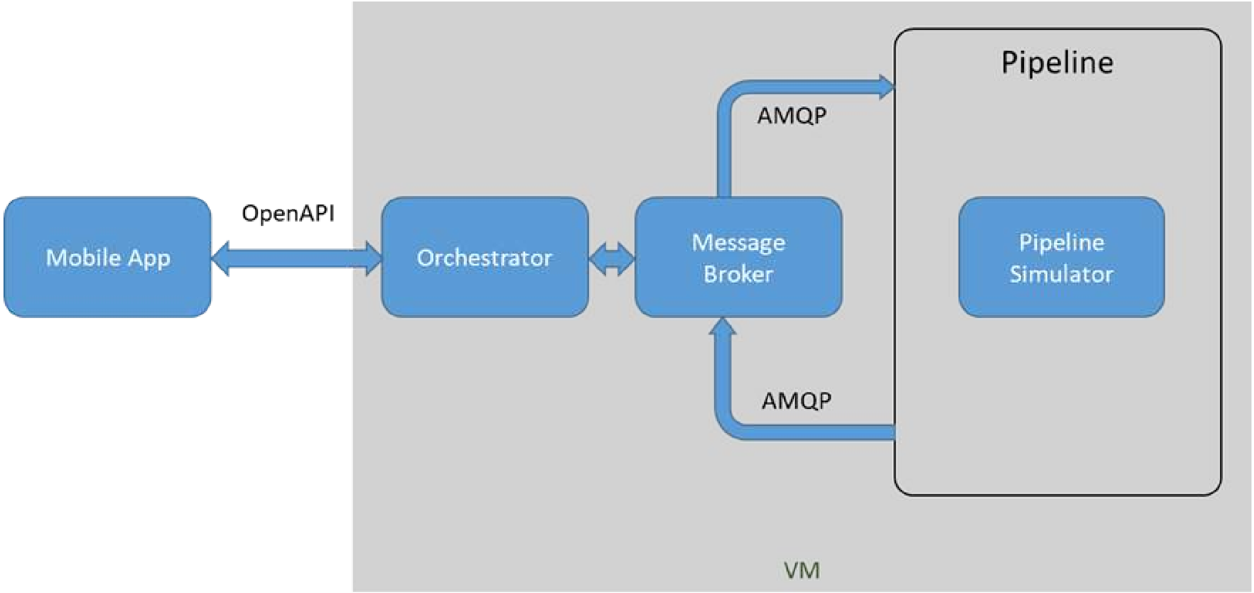


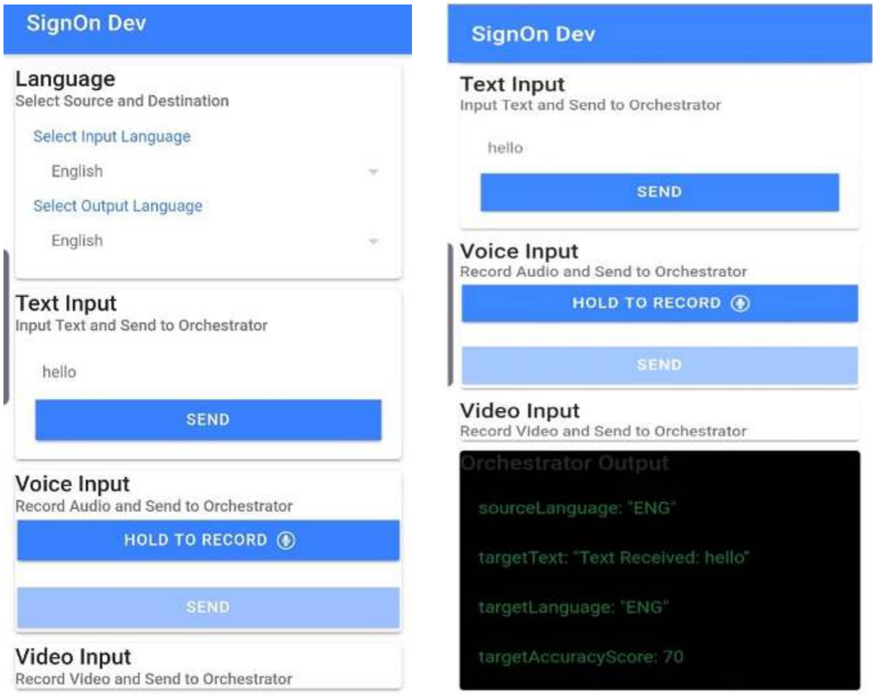
Figure 4. Initial SignON Cloud Platform

The current state of development of the cloud platform is represented by the diagram in Figure 4: an engineering development test Mobile App (signondev) has been developed and connected with the Orchestrator and Message Broker through the SignON REST API. An initial version of the Orchestrator has also been developed to demonstrate the data flow within the platform; test messages can be created with a CURL command simulating a REST call from the mobile app or using a development version of the SignON Mobile App, configured to use the initial version of the REST API. The Orchestrator then publishes the received message to a queue on the message broker, which publishes it for subscribed clients. To showcase the framework, since pipeline components are currently under development, a simulation pipeline has been devised to close the loop and send back a reply to the Mobile App. The pipeline simulator module receives the test message and echoes it sending it in output to an appropriate queue on the message broker side. The Orchestrator receives the message back and replies with a response to the received request.

This initial version of the Framework enables the setup of the necessary components to allow the pipeline components to receive messages when they are finalised, facilitating the parallelisation of the development activity.

6. SignON DevApp

An engineering development test App (signondev) is used to interact with the Orchestrator to test and verify the SignON Framework operation by accessing the evolving SignON backend services. This App is in addition to the current initial fast prototype of the users' SignON Mobile App, which is described in D6.6 (SignON Market Analysis). The SignON DevApp is available as a closed Android App (downloadable only on invitation) on the Play Store and automatically updates when SignON backend services become available.



SignOn Dev

Language
Select Source and Destination

Select Input Language: English

Select Output Language: English

Text Input
Input Text and Send to Orchestrator

hello

SEND

Voice Input
Record Audio and Send to Orchestrator

HOLD TO RECORD

SEND

Video Input
Record Video and Send to Orchestrator

SignOn Dev

Text Input
Input Text and Send to Orchestrator

hello

SEND

Voice Input
Record Audio and Send to Orchestrator

HOLD TO RECORD

SEND

Video Input
Record Video and Send to Orchestrator

```
Orchestrator Output

sourceLanguage: "ENG"

targetText: "Text Received: hello"

targetLanguage: "ENG"

targetAccuracyScore: 70
```

Figure 5. Initial SignON Development test App

7. Conclusion

The initial setup of the infrastructure (software and hardware) for the SignON application is in place and tested. The setup might change in the future due to new and evolving insights in how the different components should be arranged in order to optimize performances.