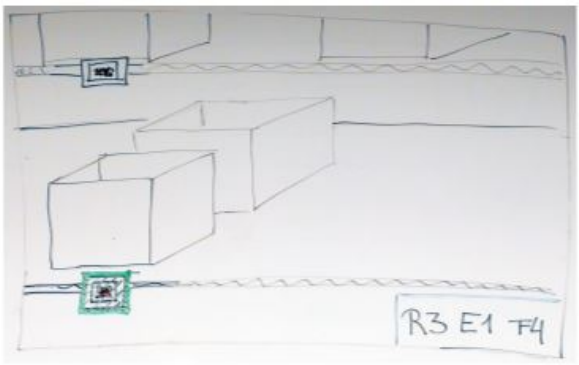
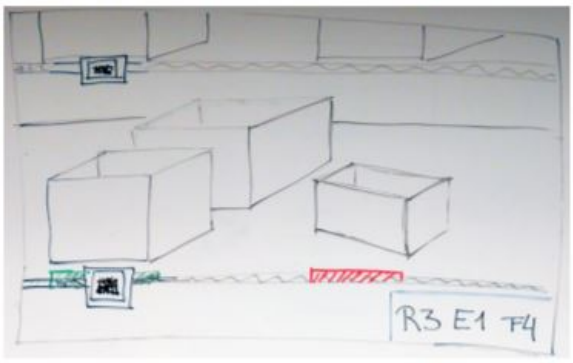
Contents

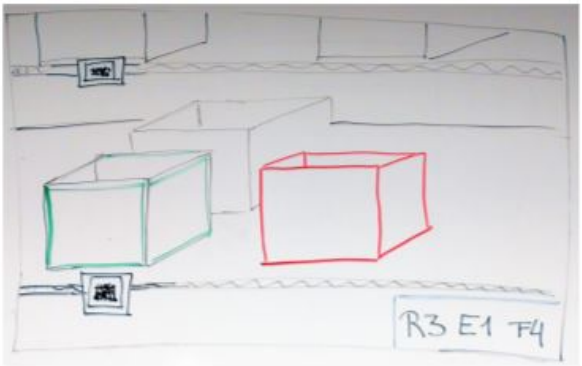
1 Product Description

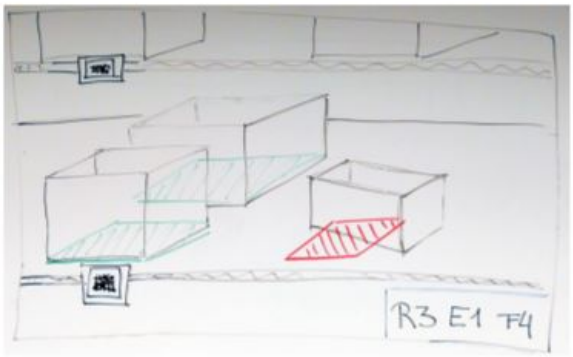
The purpose of the project is to visualize boxes that are incorrect and the pattern it belongs to in an automated warehouse using AR and the HoloLens. The Customer is the TGW group. The current situation is that a warehouse employee has to go to a stationary PC fetch the relevant data (Where are the errors located) then the employee needs to go to the correct level and shelf, has to locate the box and fix the issue. Then the issue needs to be resolved by entering data into the PC device. Resolving these issues sounds simple at the beginning but if you just know the location and box it is pretty hard if the box is missing or if the box is at the wrong position and there are no other boxes around. Then it is nearly impossible to correctly fix the issue. Therefore the HoloLens application has to display the pattern from the DB (currently displayed in green) and the error box (currently displayed in red). With this additional information it should be much easier to fix the issues. The main goal is the proof of concept and a user study to determine if the application can be used in real life (reasonable distance, dizziness, AR sickness, neck pain, ...). In order to develop a HoloLens application, a unity application needs to be created. For the AR marker tracking vuforia is used. The backend is a MySQL database with a C# REST service to communicate with the Unity-UWP app and vice versa. The typical situation is that a user takes the HoloLens, puts it on, opens the app, selects an issue from the issue list, goes to the issue location, scans the AR marker, the pattern is loaded and the error is marked with red, the employee chooses a case (box missing, not resolvable) or fixes the issue, the new state is submitted to the DB via the REST service by selecting a state in the HoloLens app, the issue list is displayed again.

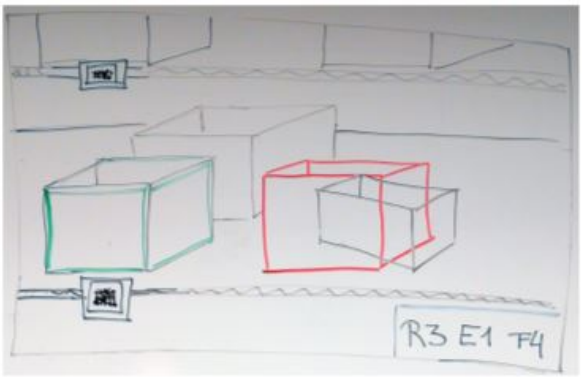
2 Visualization Concept

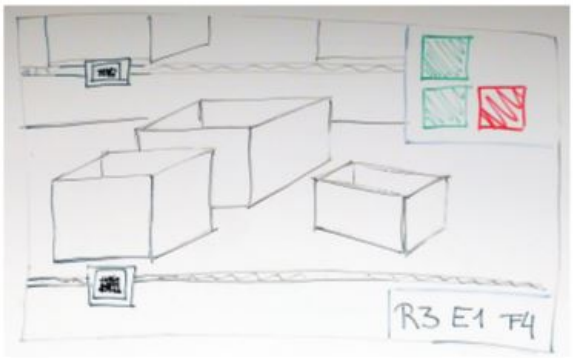


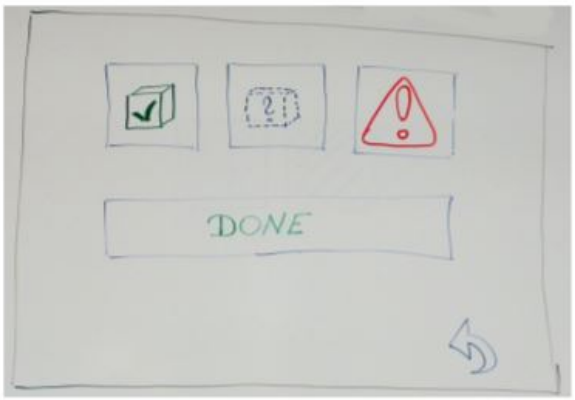










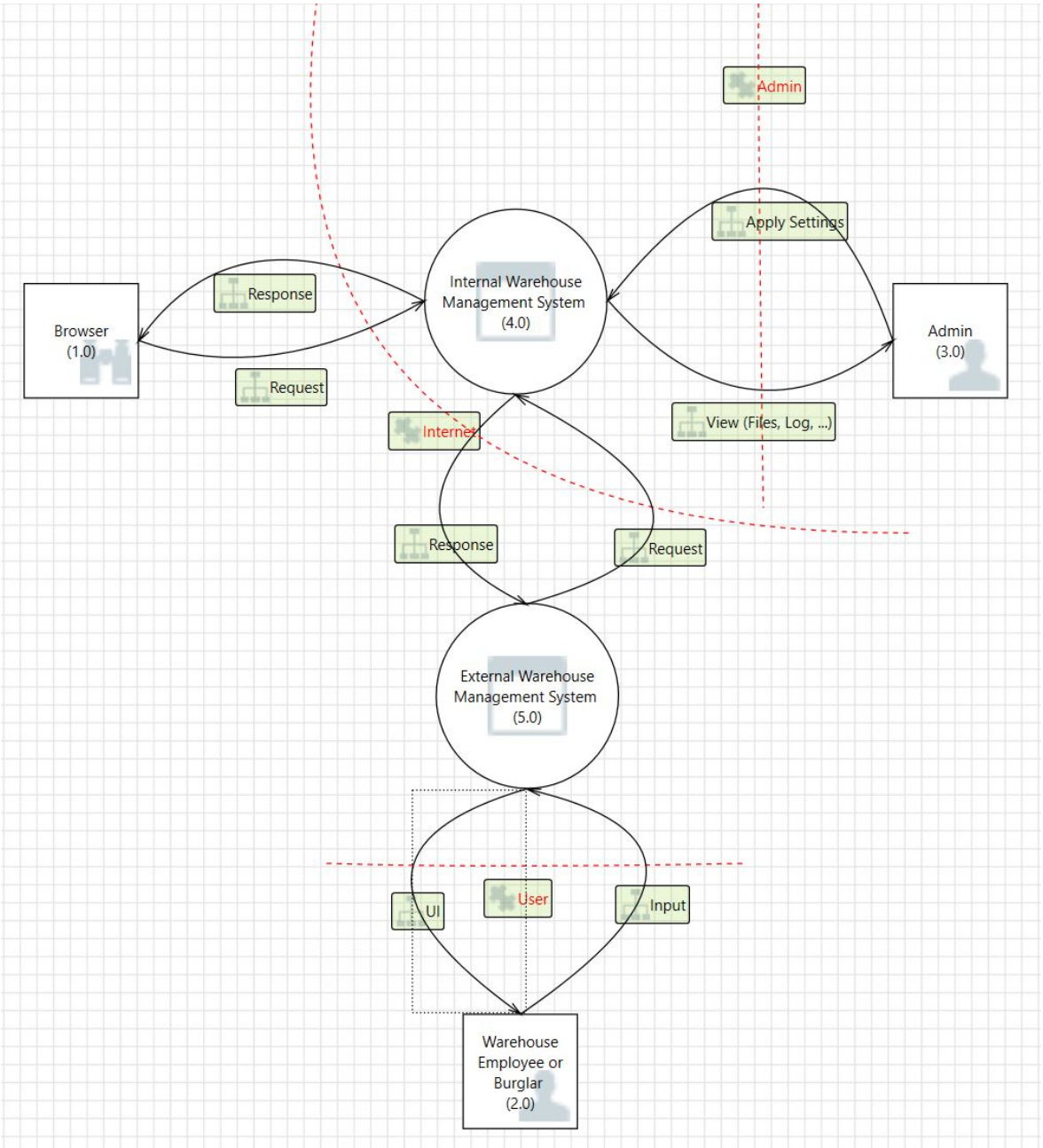


3 Threat Model

3.1 What is Threat Modeling?

Creating a model that represents the data flow in an e.g. application with all entities related to the system and the trust boundaries. This helps to visualize the process, identify threats and reduce the attack surface with counter measures at the right places.

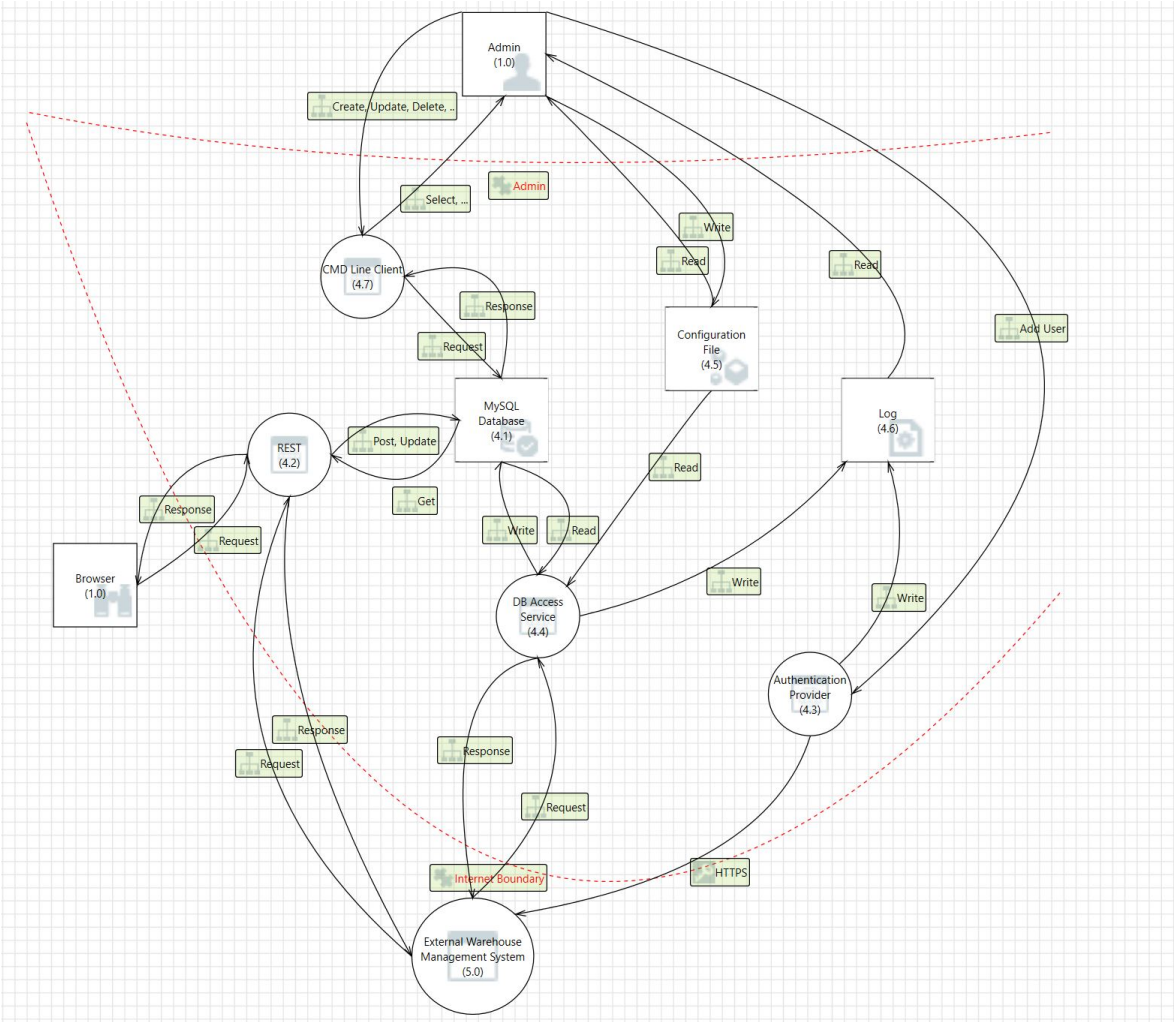
3.2 Context Diagram



Generally, the REST service is our only way to communicate with the DB. A REST service is a web service and therefore anyone can use it by default. In this specific case, the External system and a web browser can use the service. The core applications are the internal (DB, Rest) and external warehouse management systems. The external system is used by an

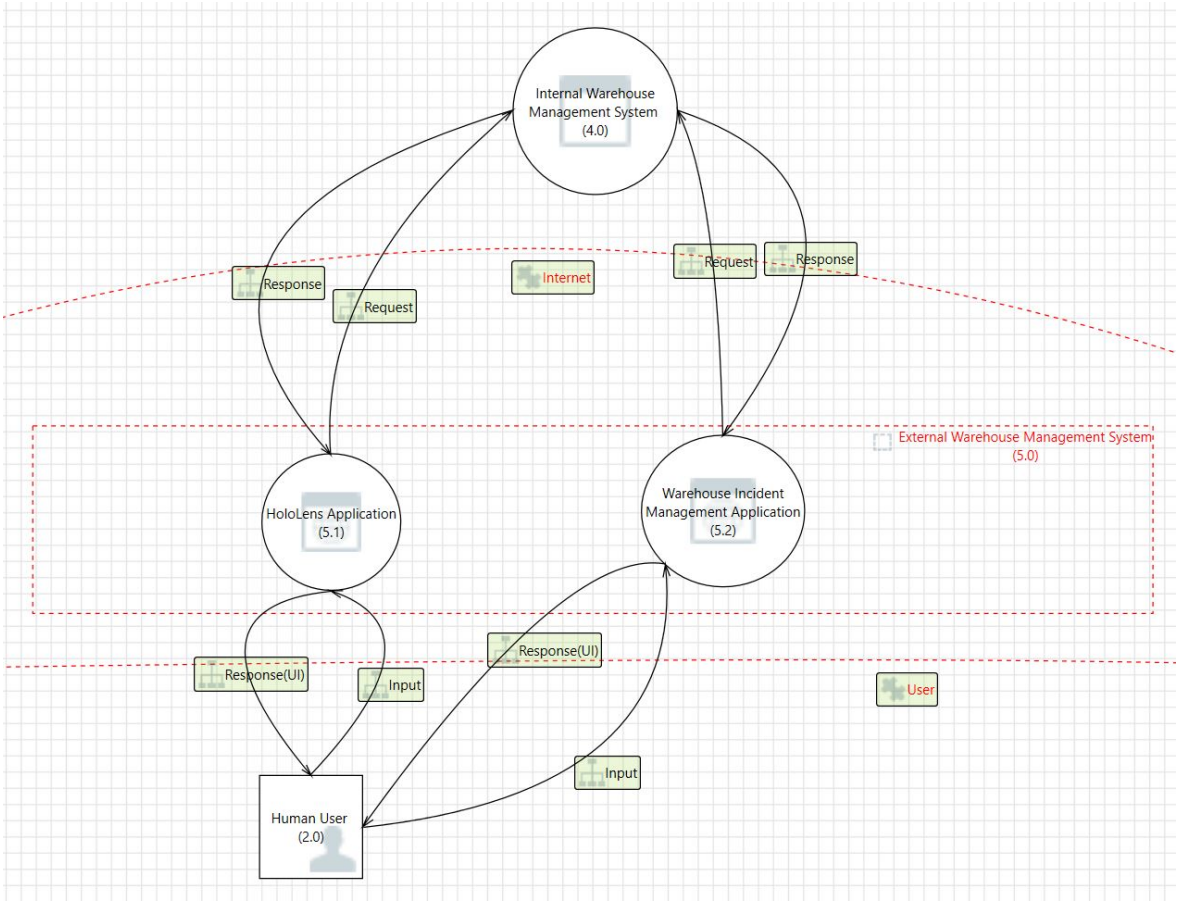
employee in the warehouse.

3.3 Level 0 Internal



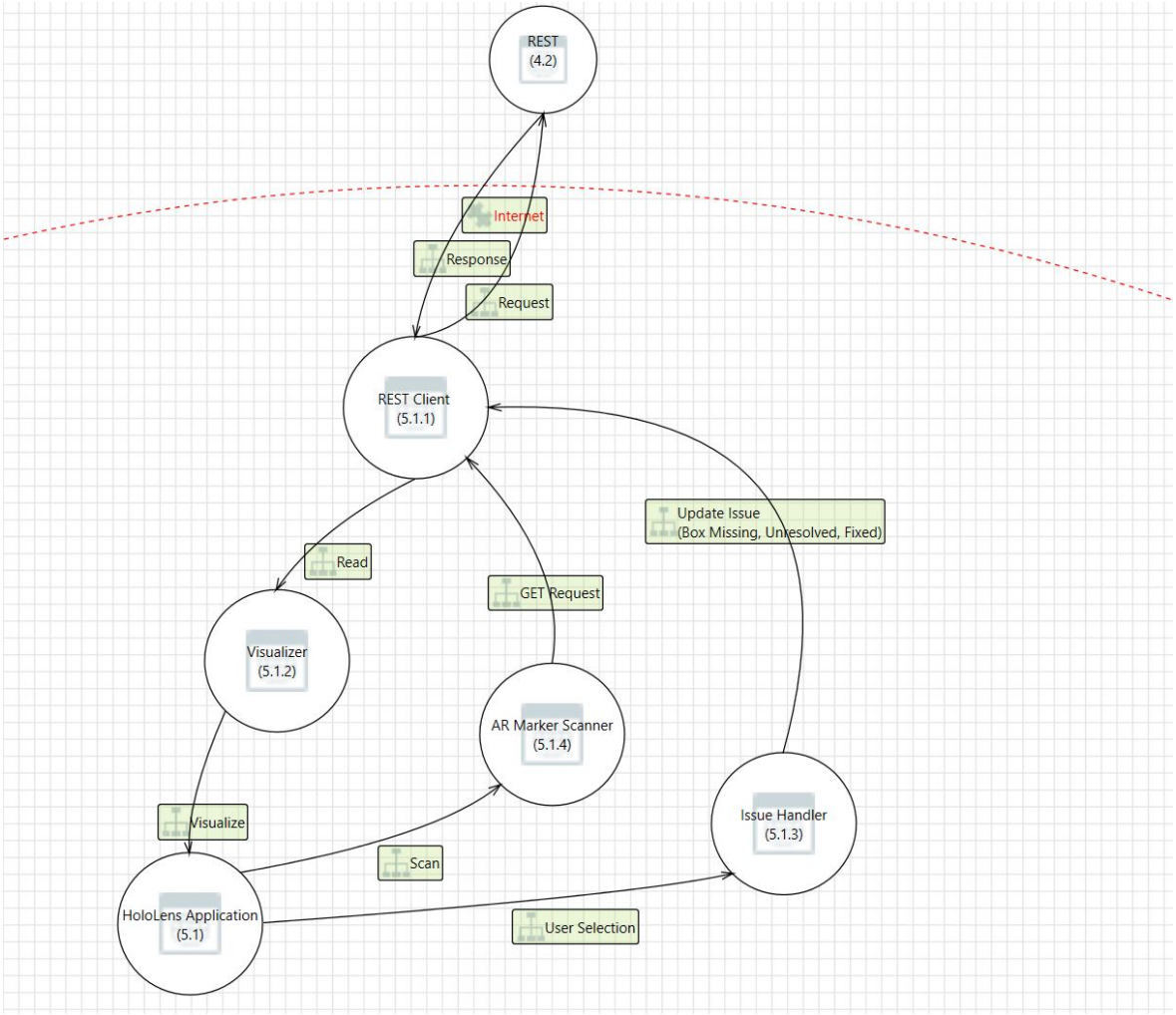
There is an existing authorization mechanism from TGW and some way to access the DB. Our part is the communication between the DB, REST service and HoloLens app.

3.4 Level 0 External



The main structure is a PC with the existing incident management app from TGW and the HoloLens app.

3.5 Level 1 HoloLens Use Case



At the start the errors are fetched via the REST service. The user selects an error from the error list. The user goes to the error location and scans the marker. The REST client fetches the data needed for the shelf and the boxes are visualized in the HoloLens app. The user tries to fix the error and submits the appropriate state via click input. The correct state is written into the DB via a REST method. The interface updates and the error list is displayed again.

4 Mitigation

4.1 Attack Points

- REST service (STRIDE, critical)
- HoloLens App (uncritical (theft, unauthorized use))

4.2 REST Authorization (Spoofing, Tampering, Information Disclosure, Denial of Service)

Authorization should be required for the REST service since it communicates directly with the DB. Nothing bad should happen if the system stays the way it is because no DELETE

operations are possible with the REST service but the employee in the warehouse could see a messed up list. This is kind of a DoS vulnerability.

- Spoofing: Authentication Provider
- Tampering: Authentication + Logging
- Repudiation: Do nothing, REST methods usually have high restrictions anyway.
- Information Disclosure: Get information with Get requests. Post response message (ID 123 not found), Authentication + Encryption + remove error message
- DoS: Caching, Scaling Docker Containers, off-line capabilities of HoloLens app.
- Elevation of Privileges: no threat for a REST service

4.3 Logging DB + REST + HoloLens App (Repudiation, Tampering)

Currently no logging is happening which is pretty bad since attacks can not be recognized.

4.4 Scalability REST (DoS

Currently, the REST service and DB are running in Docker containers but there is no option to start more than one container at the moment.

4.5 Admin Access for HoloLens App

Currently, if the product is in use at the customers warehouse the admin has no way to access any information about the application on the HoloLens but in this specific case it would be a benefit if there was a way to access information about the application.

4.6 Off-line Mode

Currently, there is no queue or a similar structure to support off-line changes.

4.7 Encrypt Communication Between REST and HoloLens App

Use HTTPS and authorization provider.