Time Simulation Extension Report

Saul van der Vies & Noël Keijzer

August 8th 2017

Total time spent: (combined)

Category 1: 8 hours 50 minutes
- Rework of 2 major methods in the pin Service to facilitate the new time system
  6 hours 15 minutes including testing and debugging
- Creation of new SystemInformationService
  45 minutes including testing and debugging
- Implementing new methods in the API Service
  50 minutes
- Rework of the Wiki
  ~1 hour

Category 2: 1 hour 25 minutes
- Looking for best way to implement the time keeping system
- Extending the test suite with basic test methods, solving minor bugs discovered while testing.

Category 3: negligible

The fact that we use asynchronous requests for our communication caused some issues with implementing the system Date requirement in the PinService as we have set up a new Service to track the system date (And in the future possibly other relevant system information).
Due to the fact that the PinService needs the system date to check if a Card is not expired it has to make an asynchronous request to the new service in the middle of the validity check for a transaction. This caused the problem that our normal error handling was no longer functional as we used Exceptions to track failure of a transaction requirement and these exceptions could not be caught outside the asynchronous receive method. It was therefore necessary to refactor the code that is responsible for handling the transactions which took some time.

Choosing to implement a new service is a big design decision.
We have chosen to create, what is at the moment a glorified time server, but could in the future be extended with other functionality such as:
- On system startup, it could communicate the location of other services to the ones that need that information, like a central hub for service to meet when necessary.
  (The service is not meant to communicate data between services that could just as well do it themselves)
- If encryption of traffic should have to be encrypted in the future, the service can function as a facilitator for the initial key exchange on system startup.

Aside from the obstacle concerning the PIN methods, implementing this extension on our system went quite easily.