

# Payment Redesign

## TECHNICAL STANDARDS IMPLEMENTATION MODEL



---

*For Next Gen API App ATMs*



7/28/2018

Version 2.0, M. Ficken

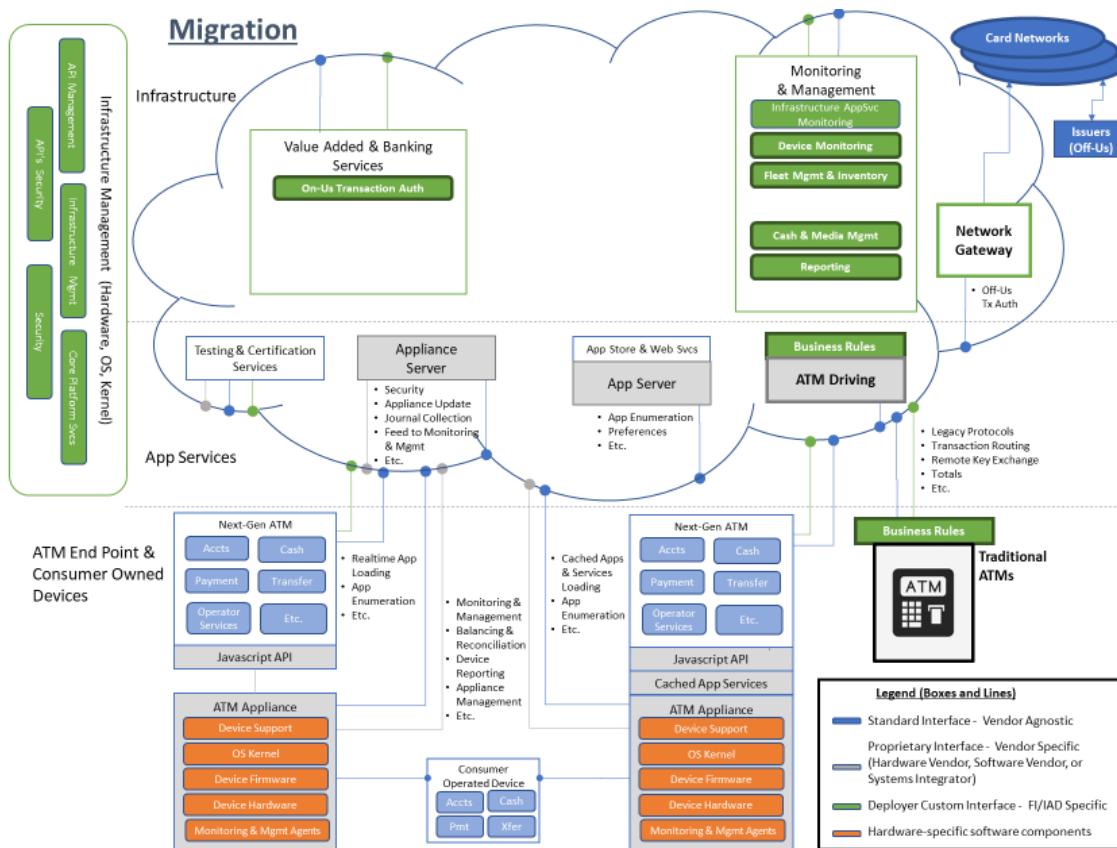
# 1 Introduction

The implementation model should define the level playing field (competition space) with freedom of development implementation choices and standardisation (co-operation space) which can be validated and certified.

Now the roadmap direction of the needed standards for NextGen are more clear, it is good to discuss the implementation model in the NextGen technical committee.

In the next paragraphs a recommendation is made to start the discussion with the aim to agree on the direction for the implementation guide and reference implementation.

The Migration Blueprint below would be the starting point for the implementation guide and first reference implementations.



## 1.1 APP-Model

Will there be an open model look-a-like mobile phones with native apps, hybrid apps, cloud apps and browser apps using thin-client app with a (cloud-based) FAT server and/or a FAT client app with a thin server or will only be one of these app models be allowed?

The JavaScript API (NextGenATMia kernel) and NextGenATM APP should preferably be device agnostic too, so it can run on ATM devices and Customer Owned mobile Devices. (Android and IOS) To create the same common user experience without additional impact, ea. integrate in the mobile banking app of the issuing bank.

To make the vendor agnostic functionality (blue) crystal clear it could be possible to allow non-vendor agnostic apps and functionality (green and orange) in the NextGen ATM APP domain. The non-vendor agnostic API can, when less competitive anymore in the future, be the input to create the future vendor agnostic NextGen API version.

### Implementation Guide

We recommend to start the implementation guide and reference implementation with an OPEN MODEL look-a-like the mobile phones, to provide the software vendors their architecture of choice, thin ATM APP client – FAT (cloud) APP server or FAT ATM APP client – Thin APP server.

## 1.2 Single / Multi development

Even for some standardisation components it is the question if every vendor should develop this (multiple developed solutions) based on the specifications or that ATMIA should provide this component as a library (single developed solution) to be integrated in the multiple developed ATM solutions.

For example, should we build one “ATMIA NextGenATM Kernel” Javascript API which can be used by all vendors to be integrated or let every vendor develop this common component.

### Implementation Guide

We recommend to start the implementation guide and reference implementation with a MULTI development model, which will be evaluated which part could become a common single developed “ATMIA NextGenATM kernel” in implementation guide version 2.0

### 1.3 Certification Service

Which implementation model for certification services will be used to provides the report with results for ATMIA NextGen approval;

- a) certifications by a 3<sup>rd</sup> party accredited Test Lab (a la EMVco)
- b) self-certification by vendor using an ATMIA certification tool (a la Visa, MasterCard)
- c) own ATMIA Test Lab service (in each region)

For any of these models to work, a comprehensive functional and technical specification is required as well a complete set of test cases and test plans.

#### Implementation Guide

We recommend to start the implementation guide and reference implementation with a SELF-CERTIFICATION model (option b) using an ATMIA certification tool. (Test Tool) vendors can build a reference implementation to be evaluated by ATMIA and later being approved by ATMIA (like MasterCard MTIP and VISA ADVT tools)

## 2 Next Steps Implementation Guide

For the Implementation Guide we recommend the following next step to create three implementation Guide worksgroups / committees with market expertise;

### 2.1 Chapter Committee – Functional Use Cases

The Basic vendor agnostic Functional Use Case Flows (end-2-end) through the blueprint components.

Committee Members (banks, deployers) should commit to invest in reference pilot at their premises

### 2.2 Chapter Committee – Technical Specifications

The Technical specification (incl. additional Javascript API layer) of the Basic Vendor Agnostic Functional Use Case Flows (see 2.1).

Committee Members (software vendors) should commit to develop reference implementation as an extension in their current CEN/XFS implementation. (for easy reference implementation)

### 2.3 Chapter Committee – Certification Services

The Certification Services for the reference implementation (see 2.2.) using a test tool based on the defined testplan and testscripts, ...

Committee Members (test tool vendors) should commit to develop a reference certification test tool implementation.