Rapid Mobile App Development Enabling Prompt Provision of New Services

AOYAGI Toru, KOBAYASHI Shigenori, KOIZUMI Ken

Abstract

This paper introduces NEC's approach to a rapid mobile application development platform and services designed to meet the era of API and "mobile first." NEC can boast of an established history in server development. In providing the "API-driven" development techniques for the rapid mobile app development platform, we intend to make the best use of our innovative technologies and proven expertise.

We are also planning to enhance API and the services that are necessary for the provision of mobile applications. The corporation will provide a development platform that is suitable for an SoE era that generates highly advanced financial services as well as savings in both time and cost. This will be achieved by API integration that will integrate the businesses that have previously been built with SoR into API, as well as by our application developments that leverage the various published APIs for implementing "open innovation".

Keywords

approach to rapid development of mobile applications, API-driven development, smart device, rapid JavaScript development, HTML5, SPA (Single Page Application), SystemDirector Enterprise.

1. Introduction

Recent financial systems have often attempted to create businesses based on a full command of the most advanced technologies or by automating or improving the efficiency of developments in the financial field.

In particular, the rapid development of digital technology has led to the creation of new services by FinTech enterprises that apply technological innovations to financial operations, etc. The release of services of financial institutions and FinTech enterprises via the API (Application Programming Interface) may also accelerate the advancement of linked financial services. In the future, it is expected that service advancements via API will expand not only in the financial sector but also in other businesses. At the same time linkages between financial businesses and other business fields may be expected via API.

Under such conditions, the application development is shifting from the traditional forms of development as performed from scratch to adoption of the API-based developments. The terminals providing the services are also shifting from PCs to the mobile terminals carried by individuals.

2. Outline of Rapid Mobile App Development and Service Provision

At NEC, we adopt mainly the following approaches to developments in the API and "mobile-first" era.

(1) Deployment of rapid development technology traditionally cultivated for the server side to the mobile domain;

(2) Enhancement of the API and services required in mobile app provisions.

Above mentioned (1) is discussed in the introduction of the development process and development platform (architecture tool) in section 3 below, and (2) is discussed in the functions related to the API and provided services in section 4.

3. Development Platform

3.1 Development Process

At NEC, we have been contributing to an early start-up of systems by means of a hybrid development process that introduces agile development in the upstream
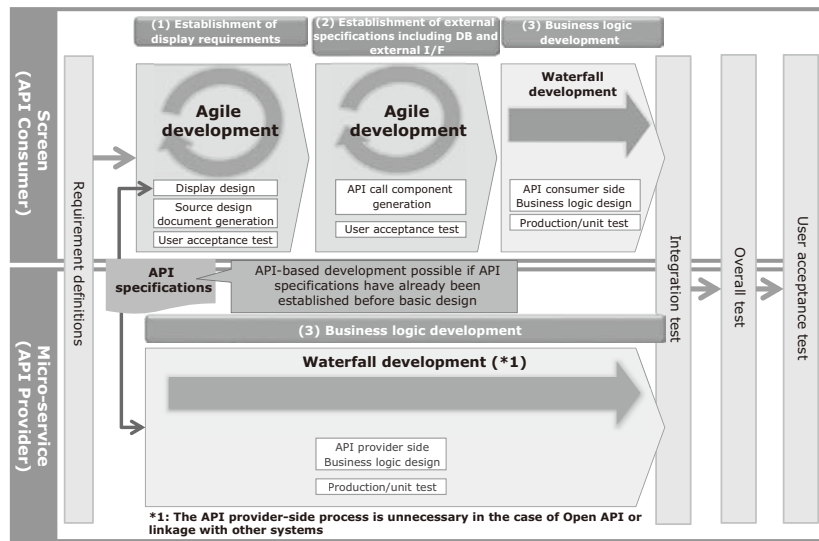
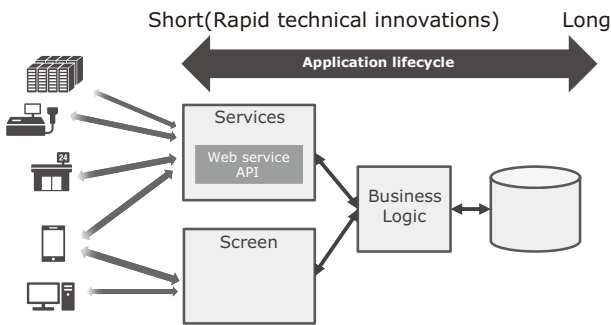Fig. 1 API-driven development process.

Fig. 2 Application lifecycle.

process where the user requirements tend to fluctuate (requirement definition and basic design).

We have advanced the hybrid development process to the "API-driven development process," with which the API consumer side and API provider side can be developed independently based on the API (**Fig. 1**).

The API-driven development process can separately develop the UI that progresses rapidly and the business logic that progresses slowly by implementing a loose coupling between the development on the API consumer side and that on the API provider side. This strategy makes possible system developments capable of long-term stable operations that focus on lifecycles (**Fig. 2**).

We are advancing the development process by taking into consideration the application of key technologies such as the Consumer-Driven Contract* and micro-service to enterprise systems.

3.2 Mobile Application Architecture

For the application architecture, we adopt the hybrid architecture, which features both the development productivity and portability of web apps and the device-specific functions of native apps (**Fig. 3**). The hybrid architecture can localize the effects of changes and thereby enable us to reduce the "Time To Market" and development/maintenance costs.

Implementation of hybrid architecture needs a mechanism capable of providing a function for bridging between web apps and native apps.

At NEC we achieve this by using a secure browser that expands the WebView components and concentrates the knowledge that has been refined via a large number of mobile projects (**Fig. 4**). The secure browser allows web apps to utilize the device-unique functions (biometric authentication and internal device linkage) via the API.

For the web apps that should be mass-produced, we adopt the Angular2 full-stack framework and the TypeScript secure coding-capable static language in order to implement robust, high-quality apps in a short delivery term and with a high productivity. In addition, the Bootstrap (CSS framework) is used for easy implementation of responsive apps that can be applied to multi-browsers and multi-devices using a single source.

It is expected that the business logic that used to be executed on the server side is going to be executed on mobile devices more frequently than at present. Therefore, we are enhancing functions such as code obfusca-

---

* Consumer-Driven Contract: Approach to the solution of problems in service integration testing in a distribution environment.
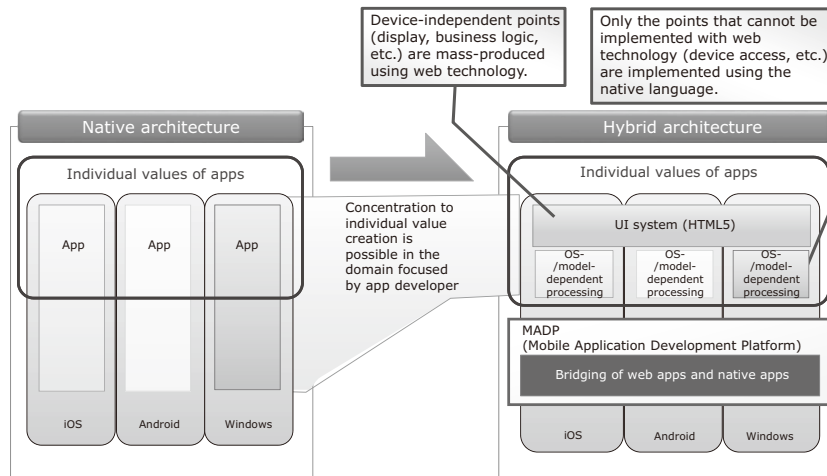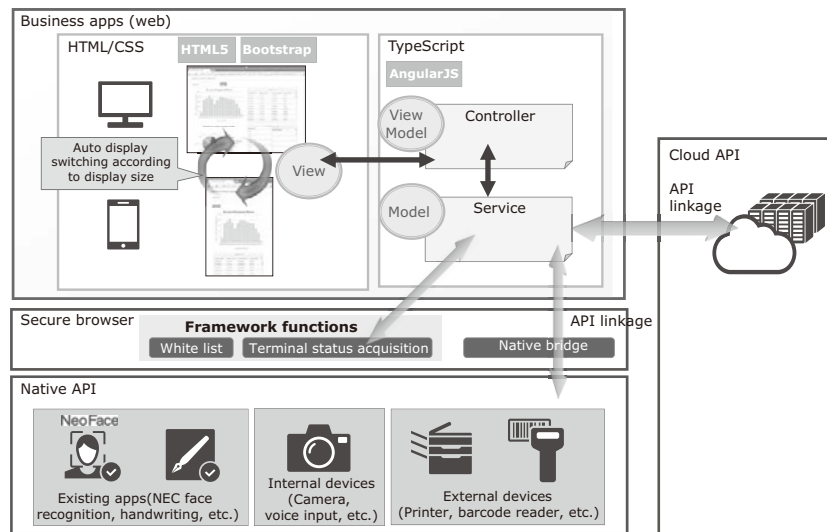
Fig. 3 Hybrid architecture.

Fig. 4 API linkage making full use of secure browsers.

tion to cope with malware and falsification in order to expand safe and secure use of the mobile applications in the financial domain.

3.3 RMAD (Rapid Mobile Application Development) Tool

Because of the rapid development of technological innovations, development in the mobile application domain necessitates a shift from the traditional coding approach using native development tools to development using RMAD tools. NEC has introduced the concept of RMAD early by means of the Real-time Development Designer (which generates the design document and web application automatically by designing the application using a WYSIWIG-based GUI) (**Fig. 5**). Backed by our rich history of achievements and the latest technologies, the Real-time Development Designer enables early implementation of value-added mobile applications that can cope with rapidly changing business environments. This is achieved in actually running the application by utilizing the API that is a corpus of business assets, while checking the UI that involves contact with customers.

The Real-time Development Designer has the following features.

(1) OpenAPI Specification to support API-driven development

For the RESTful API, the OAI (Open API Initiative) announced by Linux Foundation is conducting stan-
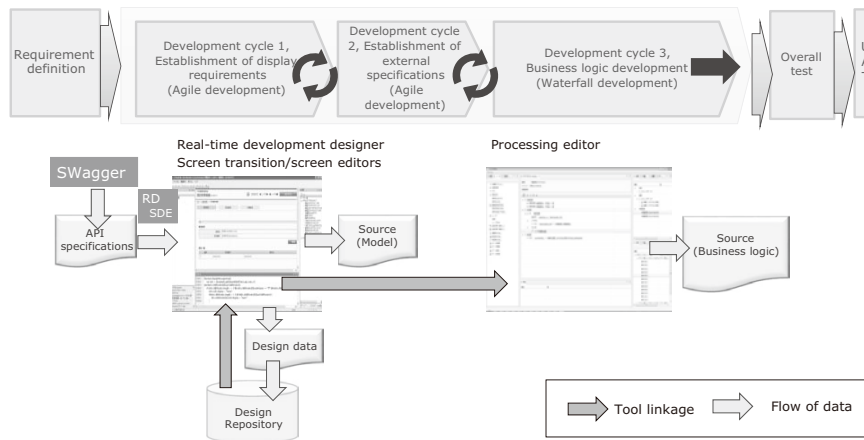
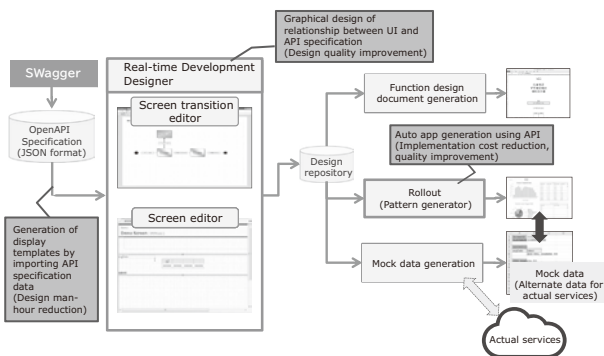Fig. 5 RMAD (Rapid Mobile Application Development) tool.

Fig. 6 Real-time Development Designer.

dardization of the OpenAPI Specification based on the Swagger Specification.

The Real-time Development Designer incorporates the API defined according to the OpenAPI 2.0 Specification and it provides the functions for automatic UI generation based on the predefined display layout (scaffolding) and linking the existing UI, API parameters and events graphically.

(2) The possibility of early specification checks by actually running an application while utilizing mock data

The Real-time Development Designer can design the SPA (Single Page Application) featuring graphical use of HTML5 using the screen transition editor and screen editor. It is also capable of responsive web design using Bootstrap and the web component function of Angular2 to enable efficient design of the UI of SPA. In addition, it provides the display input/output data in the form of Excel-format mock data so that the specifications can be checked while running the application.

(3) Generator for elimination of communication loss during process shifting, prevention of mismatching between design document and codes and the elimination of individual dependency.

After a specification check using the Real-time Development Designer, the SPA can be generated automatically using the rollout function (pattern generator) as shown in **Fig. 6**. The auto generation codes implement the MVVM (Model, View and View-Model) architecture, the Flux architecture (Publish/Subscribe type application architecture adopting single-direction data flow) and highly serviceable architectures separated per interest based on NEC's abundant SI achievements. This strategy makes it possible to develop applications with uniform quality by eliminating the individual dependency of SPA that often leads to complexity. In addition, it is also capable of generating function design documents to be used in delivery and maintenance.

4. MADP (Mobile Application Development Platform)

Development of mobile applications is required for the prompt provision of basic common components and services necessary for development by using the API in addition to the development process and tools.

4.1 Mobile Application-common APIs

The development efficiency of mobile applications is improved by receiving basic functions via API, which are commonly required for the development.

(1) SDE smart device API
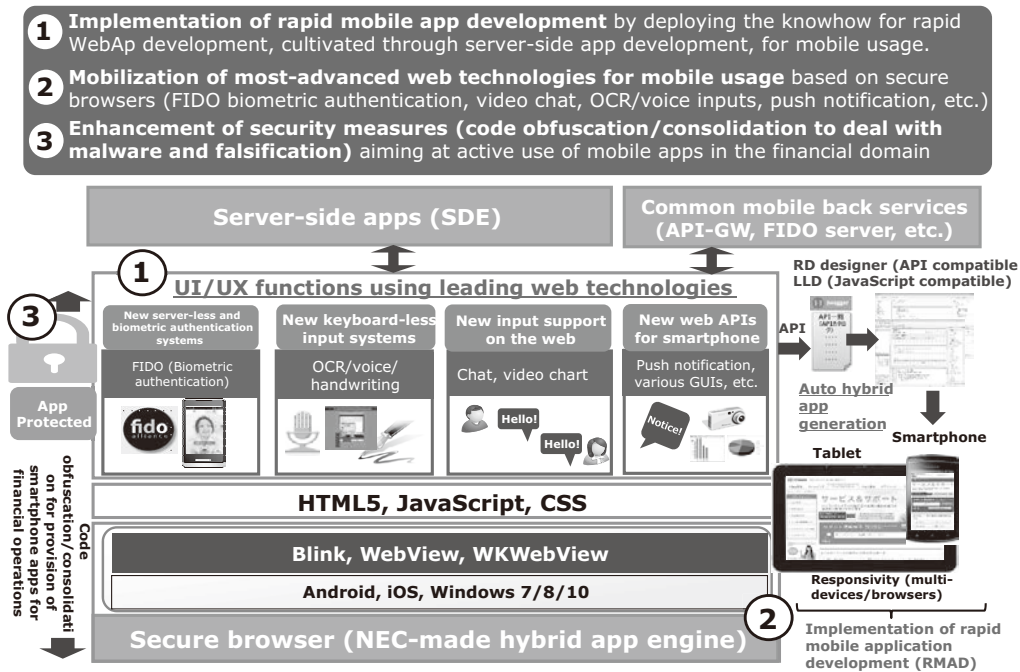
This provides the functions necessary for maximum

Fig. 7 Overall image of WebOrchestra.

utilization of the functions specific to mobile devices. The four main functions are as described below.

1) Functions enhancing the security

DB encryption, file encryption/decryption, inter-app linkage, inter-app authentication linkage, offline recognition, FIDO, linkage with biometric authentication,

2) Functions for linkage with the OS and device

Terminal status acquisition, power status acquisition, location information acquisition, camera control, printer control, barcode reader/QR code reader control, RFID radar control, beacon detection, NFC detection, sensor linkage, robot linkage, image authentication, etc.

3) Functions for linkage with the cloud

Log output, file uploading, data synchronization, push notification, server authentication linkage, terminal situation linkage (terminal status acquisition, radio wave status acquisition, location information acquisition, terminal setting), terminal remote control, file uploading, etc.

4) Enhancement of user interface

Electronic signature, smart UI (card deck, slide card, meter value input support, balloon help, file listing/selection, etc.

(2) mBaaS (mobile BaaS)

This provides server-side services for utilization of mobiles.

1) FIDO biometric authentication

A FIDO biometric authentication service is provided in addition to existing means of authentication such as OAuth 2.0 and OpenID Connect.

2) API gateway

An API gateway service linked with the FIDO biometric authentication is provided.

3) Various back services

It is also planned to provide other API services such as the push notification service, text/video chat, voice/OCR keyboard-less input services, etc.

We at NEC are also planning to provide a platform that supports the digital business service development in the SoE domain by linking various development processes, development platforms, APIs and services (tentatively named WebOrchestra) (**Fig. 7**).

5. Conclusion

In the above, we introduced our platform for mobile application development.

In addition to allowing our customers to adopt a speedy response to new deployments in their businesses that result from the acceleration of digital business, we will also support them in building an economical system that will cross barriers between enterprises by utilizing open APIs.

In the future, we will continue our efforts for the en-

hancement of mobile application developments by applying the latest technologies.

---

* Android is a trademark and/or a registered trademark of Google Inc.
* Windows and Excel are registered trademarks of Microsoft Corporation in the U.S. and other countries.
* Linux is a registered trademark or trademark of Linus Torvalds in the U.S. and other countries.
* OpenID is a registered trademark of OpenID Foundation.
* FIDO is a trademark of FIDO Alliance.
* All other proper nouns such as product names, company names, logos, etc. that appear in this paper are trademarks or registered trademarks of their respective companies.

Authors' Profiles

AOYAGI Toru
Senior Principal Engineer
Financial Systems Development Division

KOBAYASHI Shigenori
Senior Expert
Software Engineering Division

KOIZUMI Ken
Manager
Software Engineering Division

Information about the NEC Technical Journal

Thank you for reading the paper.
If you are interested in the NEC Technical Journal, you can also read other papers on our website.

Link to NEC Technical Journal website

Vol.11 No.2　FinTech That Accelerates Digital Transformation

Remarks for Special Issue on FinTech That Accelerates Digital Transformation
An Overview of NEC's FinTech Strategy

Papers for Special Issue

A New Relationship between Financing and Technology in the FinTech Era
How AI Is Transforming Financial Services
Advancing Customer Communications via AI-Robot Linkages
Safe, Reliable, Convenient Self-Monitoring Services That Use Wearable Devices
Biometrics Achieves Compatibility of Security and Convenience in Mobile Services
Rapid Mobile App Development Enabling Prompt Provision of New Services
Improvement of Financial Service Safety by Promoting Cyber Security Measures
Enhancing FinTech Security with Secure Multi-Party Computation Technology

NEC Information

NEWS
2016 C&C Prize Ceremony

NEC Technical Journal
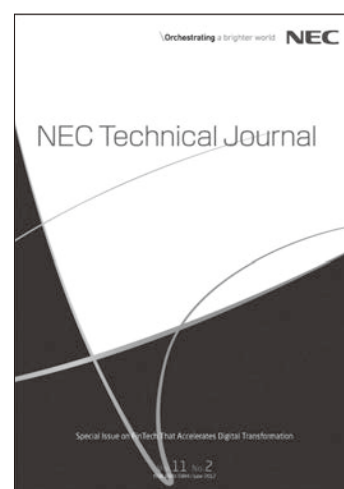
Vol.11 No.2
June 2017

Special Issue TOP