



gemalto
security to be free

JavaOne

Web 2.0 Applications on a Next-Generation Java Card™ Platform

Laurent Lagosanto
Jean-Jacques Vandewalle

Research Engineers
Gemalto

<http://www.gemalto.com>

TS-5203

Goal of This Talk

Learn how to build a Web Application with a Next-Generation Java Card™ platform

Understand the value of Next-Generation Java Card technology-based Web applications (Java Card Web applications) in the context of Web 2.0

Agenda

Next-Generation Java Card Platform

NG Java Card Technology and Web 2.0

Demo: A NG Java Card technology-based
Web 2.0 PIM

Under the hood of the demo

Conclusion and perspectives

Agenda

Next-Generation Java Card Platform

NG Java Card Technology and Web 2.0

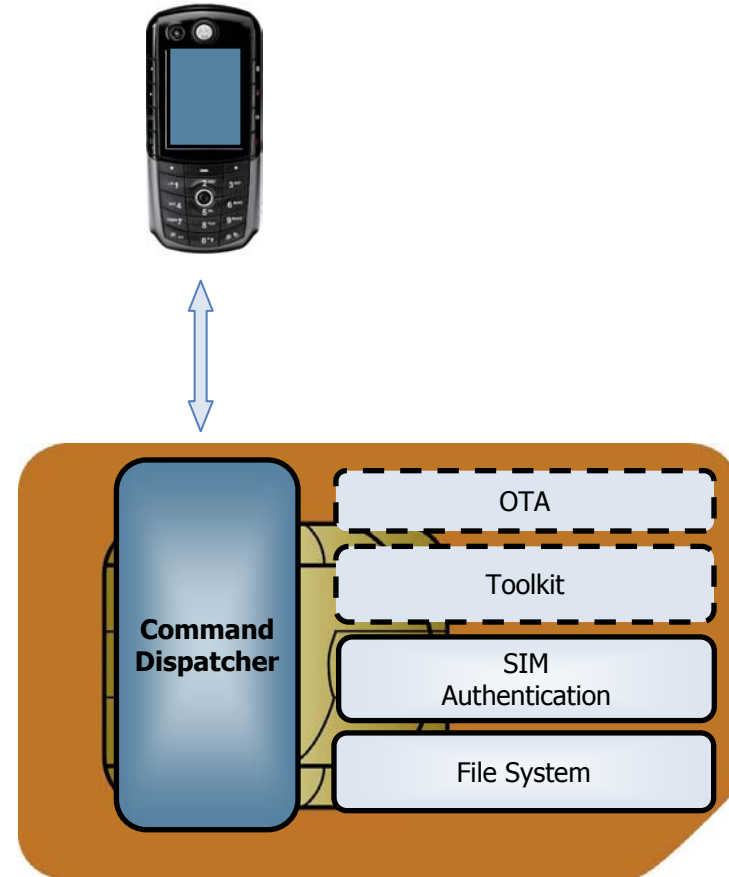
Demo: A NG Java Card technology-based
Web 2.0 PIM

Under the hood of the demo

Conclusion and perspectives

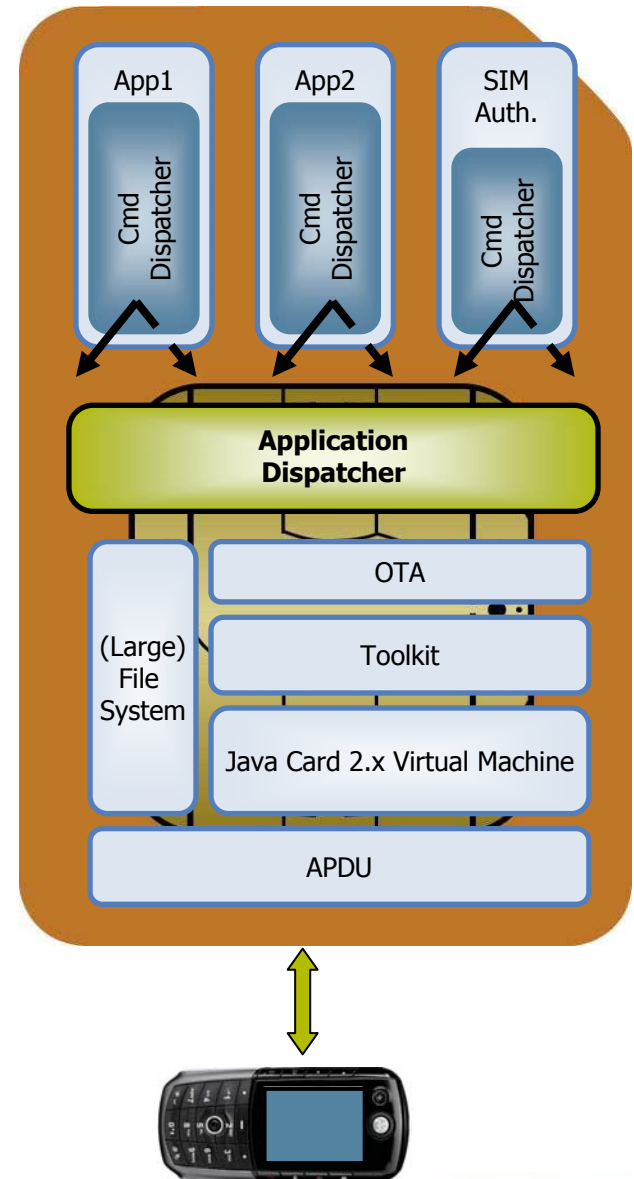
90's: Proprietary OS Cards

- 1st generation of SIM cards
 - OS architecture = applicative command dispatcher + authentication module + file system
- Major evolutions
 - Proprietary interpreter: 1st step of toolkit services
 - Over the Air (OTA) access: way for operators to manage cards remotely
- Market driven by the boom of mobile communications



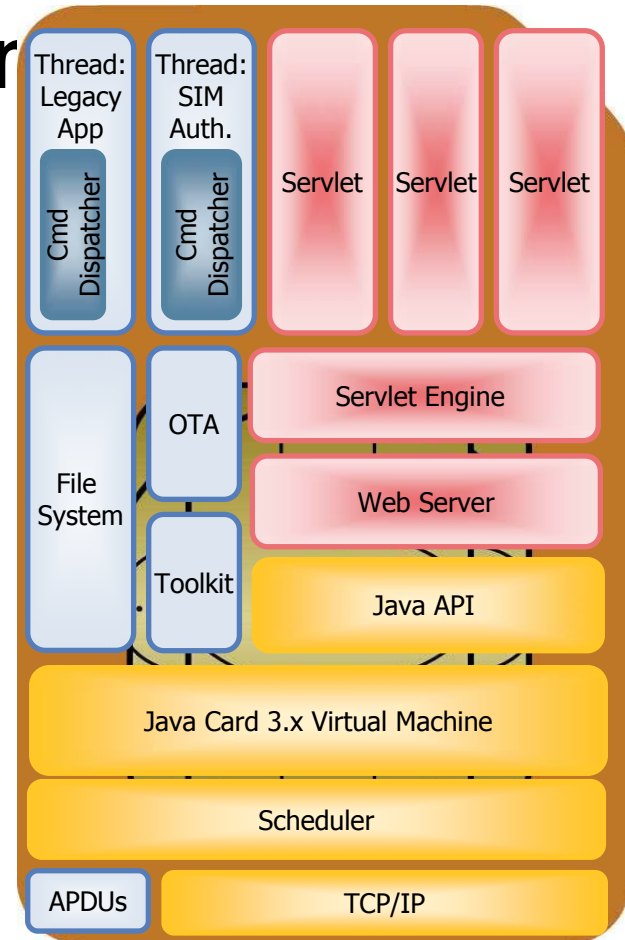
Java Card Platform: Open OS Era

- **2nd generation of SIM cards**
 - OS architecture = dispatching to applets running on top of Java Card Virtual Machine (VM)
- Key benefits
 - Interoperability across card manufacturers
 - Post-issuance downloading
- Major evolutions
 - Interoperability
- Market driven by demand for **interoperability and memory capacity**



Next-Generation Java Card Platform: A Connected Platform

- 3rd generation of SIM cards
 - OS architecture = multithreaded Java™ Virtual Machine (JVM™) + TCP/IP + general purpose APIs
- Key benefits
 - New application model: Web Apps
 - Network aware (TCP/IP)
- Major evolutions
 - USB high-speed protocol
 - NAND Flash mass storage



NG Java Card is still a work-in-progress, not final specifications.
The terms “Java Virtual Machine” and “JVM” mean a Virtual Machine for the Java™ platform.

Next-Generation Java Card Platform

for Java Card Developers

- Backward compatibility
 - `javacard.framework` is still supported
 - Applets and APDUs are still supported
 - APDUs are messages of the legacy smartcard application protocol
 - Applets are Java Card 2.x instances for smartcard applications
 - Same memory model (with an automatic GC)
 - The firewall is still enforcing the security rules
- New features
 - Multithreading, String, long
 - TCP/IP connectivity, abstracted by streams
 - Web application support: HTTP Servlets in addition of Applets, to dynamically extend the embedded web server behavior

NG Java Card is still a work-in-progress, not final specifications

Next-Generation Java Card Platform

for Web Developers

- Well-known application model
 - Servlet 2.4 API subset
 - Applications are packaged in .jar files (classes and files)
 - CLDC Generic Connection Framework (GCF)
 - SSL/TLS capable
- Additions and enhancements
 - Dedicated crypto API
 - Firewall enforces isolation between applications
 - Persistent operating system: the Java Card VM never stops
- New features
 - New descriptors, to manage new deployment and security aspects
 - The smallest server you can imagine ;-)

NG Java Card is still a work-in-progress, not final specifications

Agenda

Next-Generation Java Card Platform

NG Java Card Technology and Web 2.0

Demo: A NG Java Card technology-based
Web 2.0 PIM

Under the hood of the demo

Conclusion and perspectives

Web and Java Card Technology

More and more convergence

The Web

Before 2000:

- 90's: Rapid deployment
- 1995: Java technology Applet for the Web
 - Secure mobile code for the Web
- 2000: Internet bubble (Web 1.0)

Since 2000:

- Web everywhere for everything: DSL, WiFi, 3G, VPNs, VoIP,...
- Plethora of smart technologies: FF, CSS, DOM, JavaScript™ technology (AJAX), RSS, XML, SOAP (Web Services)
- **Web 2.0: Richer applications, WebMail, blog, calendar, Social Web sites, Mashups**

The Smart Card

Before 2000:

- 90's: worldwide deployment
- 1997: Java Card Applet for Cards
 - Secure code for smartcards
- 2000: WAP for mobile phones

Since 2000:

- Java Card™ 2.x SIM mass deployments
- Many applications: large SIM cards, EMV Migration, emerging security and ID businesses
- **Novelties: larger memories, USB, TCP/IP, Next-Generation Java Card platform with Servlet engine**

Web 2.0 and Next-Generation Java Card Platform

The right time to meet

- Web 2.0: a more ubiquitous Web than ever
 - Desktop **applications are moving to the Web**
 - Richer and dynamic Web applications (**AJAX**)
 - Web applications **on mobile phones**
- NG Java Card platform: embeddable Web applications
 - **TCP/IP + HTTP(S)** over high-speed protocol (**USB**)
 - Static (files) and dynamic (Servlet) content served by an **embedded Servlet engine** with mass-storage available
 - **Security enabler** for Web applications
 - **Personal web server opportunities**

Web 2.0 and NG Java Card Platform: Opportunity #1

Beyond SIM Toolkit menus

- Thanks to the embedded HTTP server, pages will be provided to the mobile browser
 - Text-based SIM Toolkit menus replaced by rich HTML-based GUI

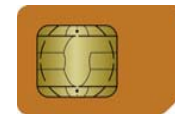


SIM application Toolkit is an API used to create card applications that can display text and menus on the phone screen

Web 2.0 and NG Java Card Platform: Opportunity #2

Local Web applications

- NG Java Card platform = a **local** Web server serving operator's Web applications
 - The more capable the browser in the phone, the richer the card Web applications
 - Support for CSS, JavaScript technology,
 - DOM, XHR, etc
 - Fast and secure access to on-card content
 - An alternative to remote server hosting (personal) content
 - Runs offline



Web 2.0 and NG Java Card Platform: Opportunity #3

Connected local Web applications

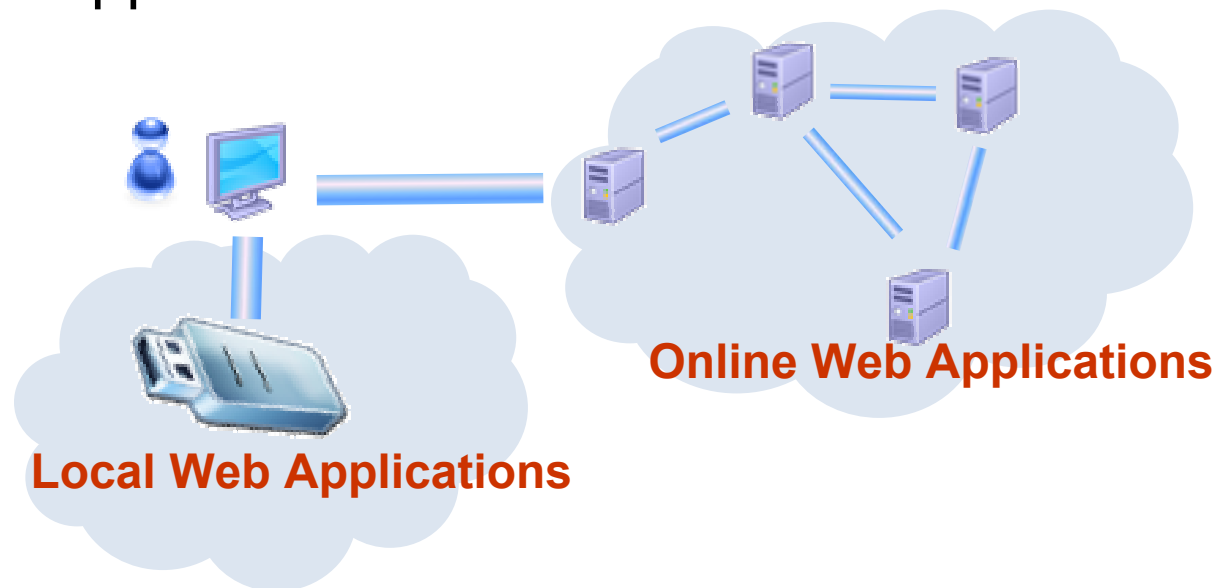
- Local Web applications can mashup with online Web applications
 - Local applications **enriched** with online content
 - Online applications **personalized** with local content

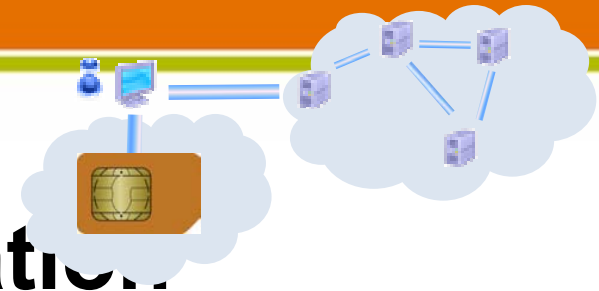


Web 2.0 and NG Java Card Platform: opportunity #4

Beyond SIM cards

- Embedding NG Java Card technology into a smart USB Flash Drive
 - Smartcard secure microcontroller + Flash memories
 - Serving multimedia data, rich personal and secure Web applications





A New Kind of Application

Serving Web applications from a NG Java Card technology-based device (Java Card device)

- Allow to make **personal** Web applications **portable** across system and devices
- Maintain **privacy** of personal data in a **controlled** local Web server
- Benefit of **richer** Web applications look and feel
- Work even if **offline**
- NG Java Card platform local Web applications can be **mixed** with online services using Web **mashup** techniques
- NG Java Card platform local Web server is remotely **manageable**

Building a NG Java Card Web 2.0 Application

Technical architecture

- Characteristics of NG Java Card devices
 - Limited processing power, bandwidth and memory space
 - Web 1.0 architecture not relevant: involves important resource utilization at each page refresh
- Making rich NG Java Card Web 2.0 applications
 - A 3-tier architecture
 - AJAX development methodology
 - Web Mashup capabilities

Building a NG Java Card Web 2.0 Application

Leveraging on a 3-Tier architecture



Presentation

Initial rendering and behavior

Runs JavaScript programming language upon user actions

`XMLHttpRequest`(Get/Update/Del data)

Runs JavaScript programming language to decode and inject data into the page
(DOM object creation or `innerHTML`)



Application

JavaScript technology, CSS, HTML, images static files

Runs business logic in Servlet

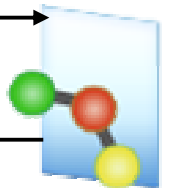
Returns data encoded

Data



Resources

Accesses data

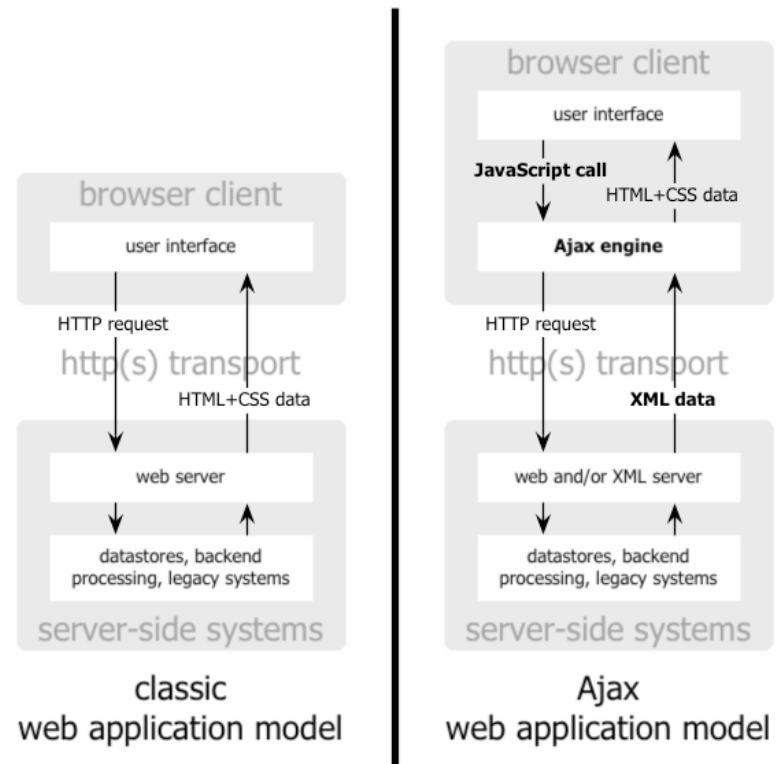


POJOs in Java Card VM persistent heap

Building a NG Java Card Web 2.0 Application

AJAX = a set of browser technologies

- HTML and CSS for presenting
- JavaScript technology for local processing
- DOM (Document Object Model) to access data inside the page
- **XMLHttpRequest** for asynchronous data retrieval

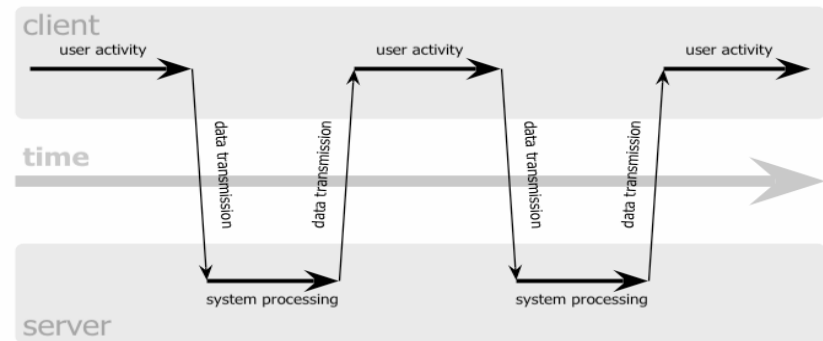


Building a NG Java Card Web 2.0 Application

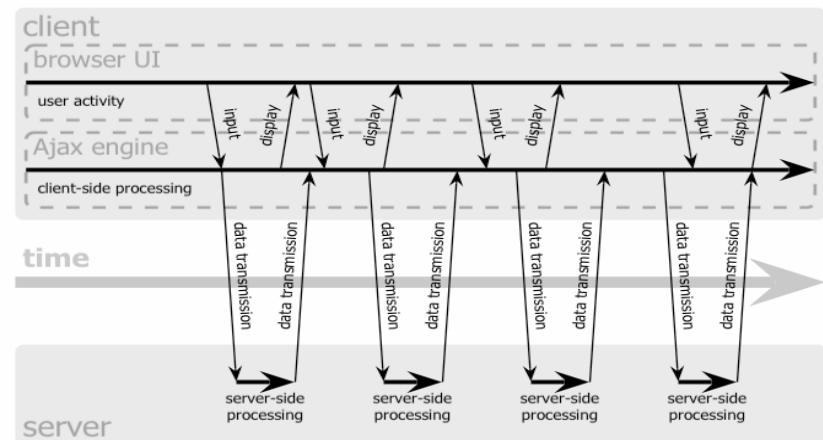
XMLHttpRequest = no more single request/response

- Sends and reads data on the server asynchronously
 - Flexibility to get the data from the server only when needed
 - Flexibility to get only the needed data (and no more)
- Asynchronous mode means
 - Server response processed when available
 - No need to wait and to freeze the display of the page

classic web application model (synchronous)



Ajax web application model (asynchronous)



Source: <http://www.adaptivepath.com/publications/essays/archives/000385.php>

Building a NG Java Card Web 2.0 Application

Leveraging on AJAX development methodology

- To build a fast and dynamic website
 - Selective update of page elements (JavaScript technology+DOM) with data provided by the server (XHR)
 - No whole page reload required (XHR)
- To save resources by using the power of the client browsers
 - Processing on client (JavaScript)
 - Small data provided by the server only when needed (XHR)
- **Intensively use AJAX to save NG Java Card platform server resources while developing fast and dynamic applications**

Building a NG Java Card Web 2.0 Application

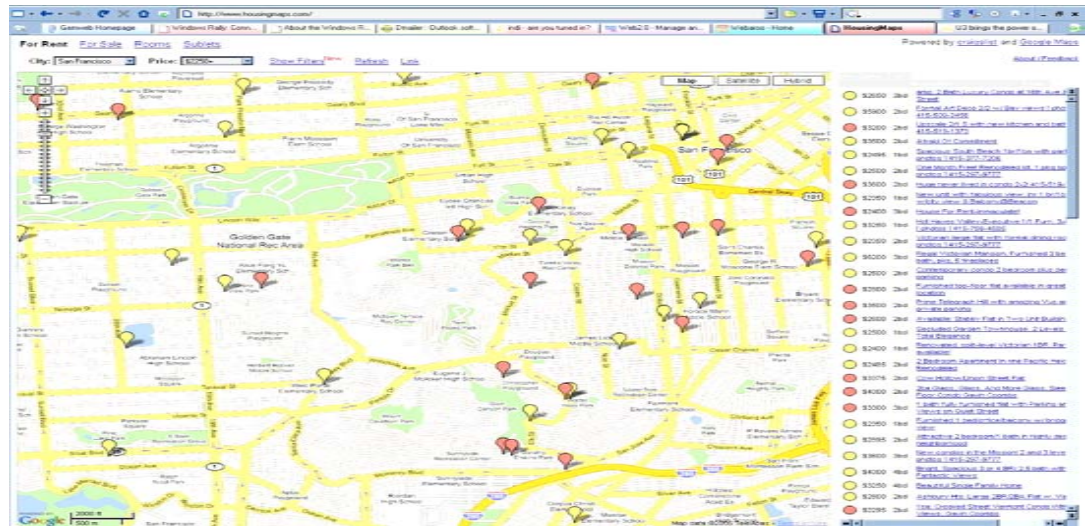
Web mashup capabilities

- “A mashup is a Web application that uses content from more than one source to create a new service”
Source: Wikipedia -- [http://en.wikipedia.org/wiki/Mashup_\(web_application_hybrid\)](http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid))
- Relies on public Web service API
 - Google Maps, Yahoo! Maps, Amazon, Flickr, del.icio.us, etc.
- New applications created by reusing existing services

Source: New Scientist (2006-05-12)



Housingmap.com

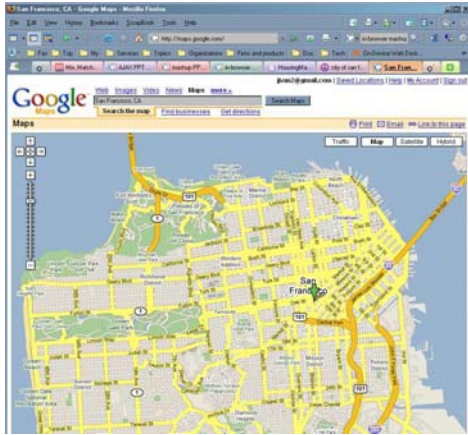




JavaOne

Building a NG Java Card Web 2.0 Application

Mashup implementation



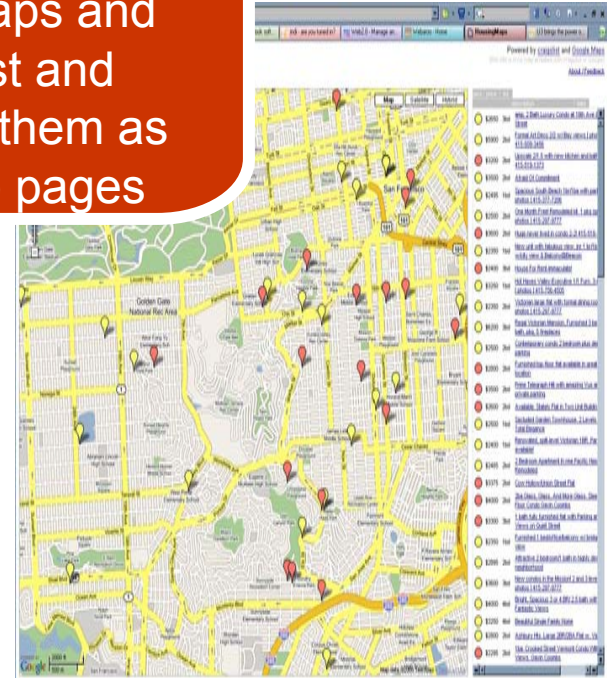
From GoogleMaps



From CraigList



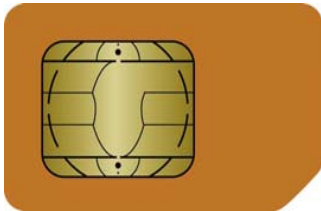
Housingmap.com server gets data from GoogleMaps and CraigList and assemble them as new Web pages



Housingmap.com

Building a NG Java Card Web 2.0 Application

Mashup with a NG Java Card Web application: in an external server



From NG Java Card device

From elsewhere



Requires an external server accessing to personal content

- External Web server and mash-up code deployment issues
- External Web server with read access on personal data



Resulting page



Building a NG Java Card Web 2.0 Application

Mashup with a NG Java Card Web application: in NG Java Card device

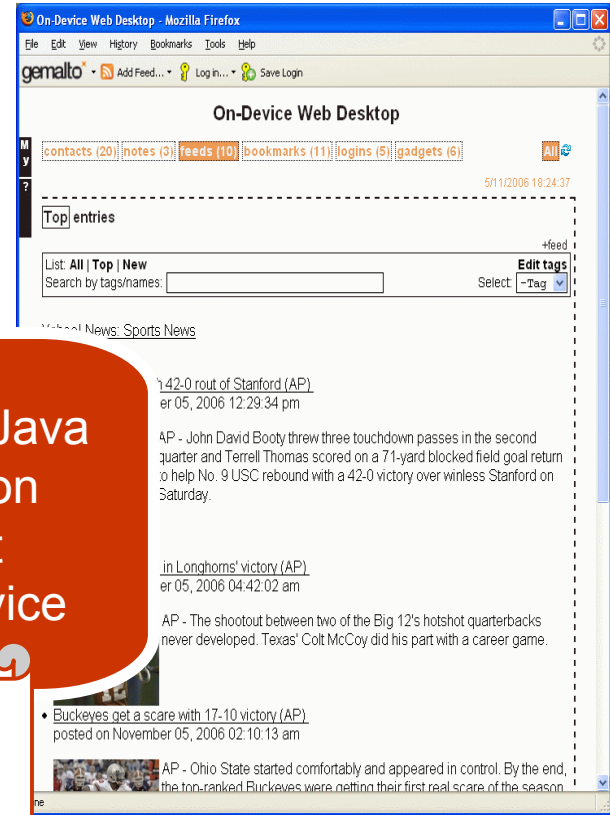


From elsewhere



Mashing-up from NG Java Card Web application requires important processing in the device

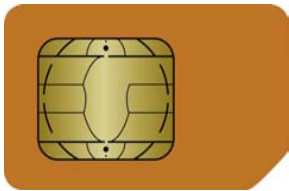
- May be limited by NG Java Card device bandwidth
- May be limited by NG Java Card device capabilities (CPU, RAM, etc.)



Resulting page

Building a NG Java Card Web 2.0 Application

Mashup with a NG Java Card Web application: in-browser

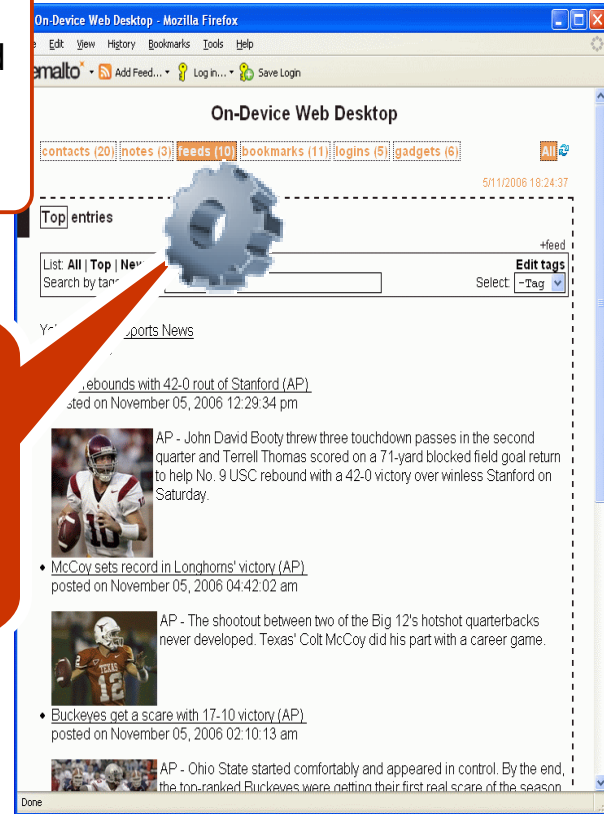


- No deployment nor security issues: all code come from the NG Java Card device
- Not limited by NG Java Card device capabilities

Mashing-up inside the browser from the NG Java Card device provided JavaScript technology:

- gets data from elsewhere
- injects data to the page

From elsewhere



Resulting page



Building a NG Java Card Web 2.0 Application

Leveraging on Web mashup capabilities

- Best solution = “in browser” mashup
 - Self-contained and safe: does not require an external server
 - Runs in the browser: saves NG Java Card platform server resources
- Issue with the “in-browser” mashup solution
 - Browser prevents XHR calls from outside the origin domain of the page (aka “same origin policy”)
 - NG Java Card device-originated JavaScript technology codes cannot request content from “elsewhere”
- Solution: use of external content formatted as JavaScript technology code (a.k.a. “on-demand JavaScript” technology)
 - Loading dynamically JavaScript technology code from another domain is not forbidden
 - Web ads use intensively this technique
- **Intensively use on-demand JavaScript programming language technique to save NG Java Card platform server resources while enriching the application with mashups**

Building a NG Java Card Web 2.0 Application

What makes an NG Java Card application “Web 2.0”!

- Design a **3-tier architecture** to embed only the vital part of the application
 - Business logic in Servlet codes
 - Persistent data for free with NG Java Card technology memory model
- Use **AJAX** to save resources on the NG Java Card platform server
 - HTML pages don't need to be generated by Servlet codes
 - With **XMLHttpRequest**, limit exchanges to on-user-demand need for data
 - Use simple data formats: plain text, XML, JSON, etc.
- Use **AJAX** to make the Web application responsive and dynamic
 - **XMLHttpRequest** prevents the pages from freezing
 - JavaScript technology and DOM allow for limited page updates
- Enrich your application with online content
 - Prefer **in-browser mashup** to save server resources

Agenda

Next-Generation Java Card Platform

NG Java Card Technology and Web 2.0

**Demo: A NG Java Card
technology-based Web 2.0 PIM**

Under the hood of the demo

Conclusion and perspectives

The demo is a prototype anticipating some NG Java Card technology-based functionalities

NG Java Card Technology-Based Web 2.0 PIM

An embedded Web Personal Information Manager

- User's personal information



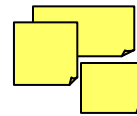
Contacts



News



Bookmarks



Notes



Login accounts

...

- On a personal device



- Accessible from any host (online or offline)



The demo is a prototype anticipating some NG Java Card technology-based functionalities

NG Java Card Technology-Based Web PIM

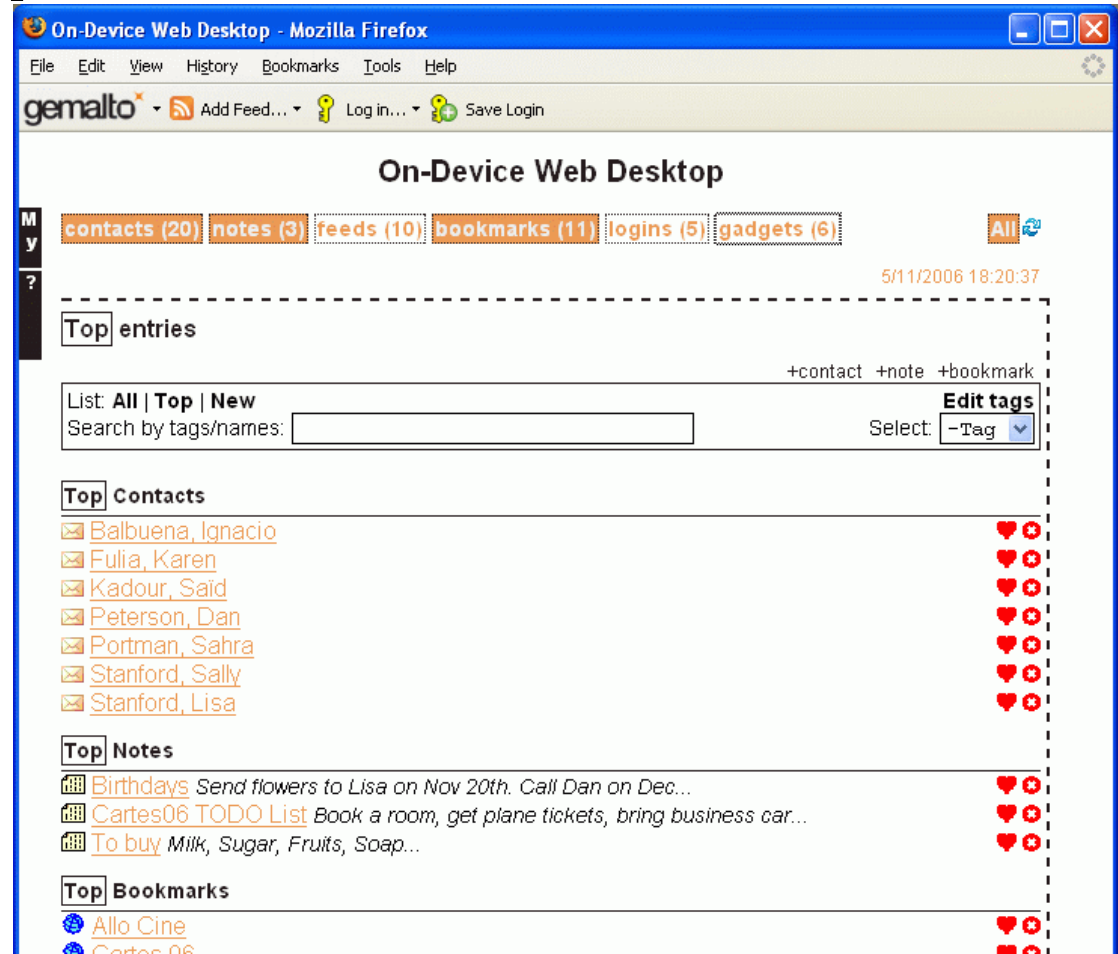
Application features

- Rich user experience: use of AJAX for better responsiveness
- Rich content: mashup of information from different sources
- Ubiquity: available offline
- Privacy: personal data is under user's control
- Security: login/password management and anti-phishing solution

The demo is a prototype anticipating some NG Java Card technology-based functionalities

NG Java Card Technology-Based Web PIM

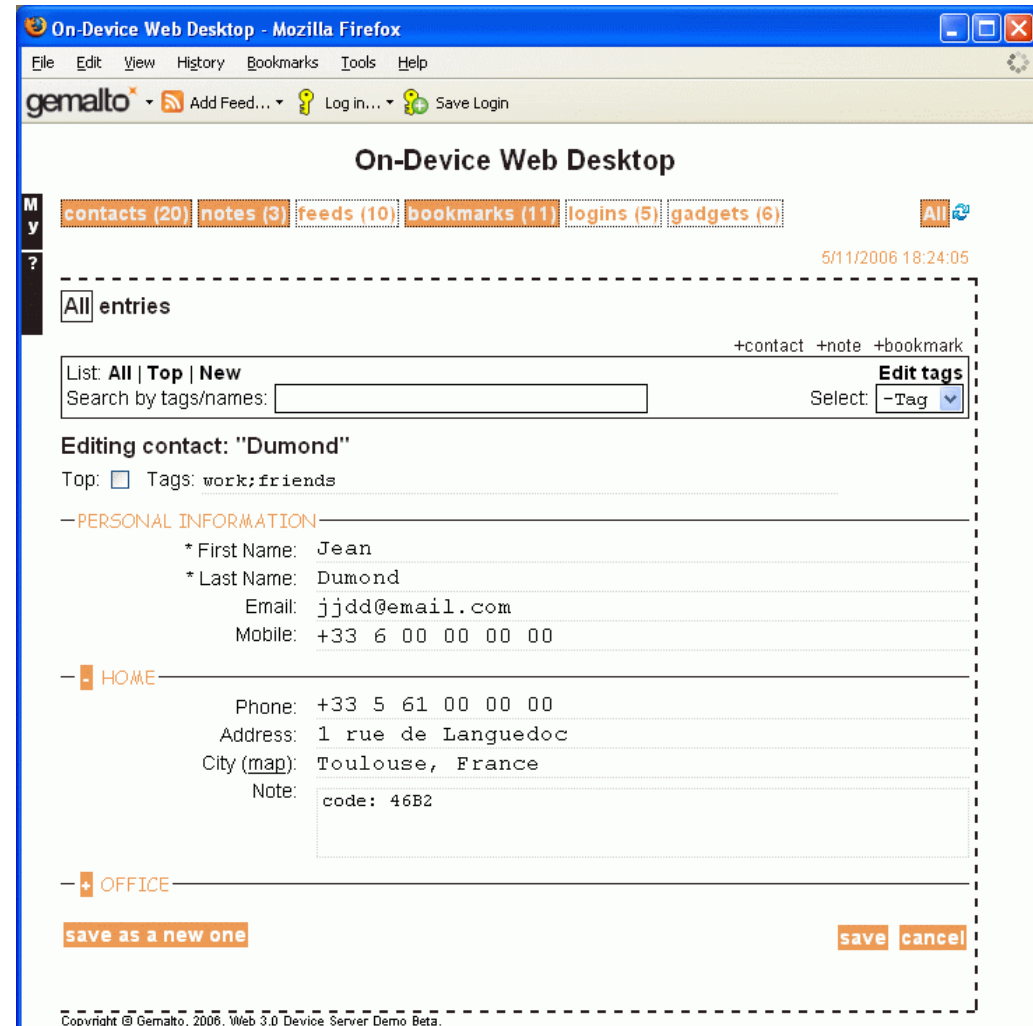
User's data GUI



The demo is a prototype anticipating some NG Java Card technology-based functionalities

NG Java Card Technology-Based Web PIM

User-friendly interface to edit personal data



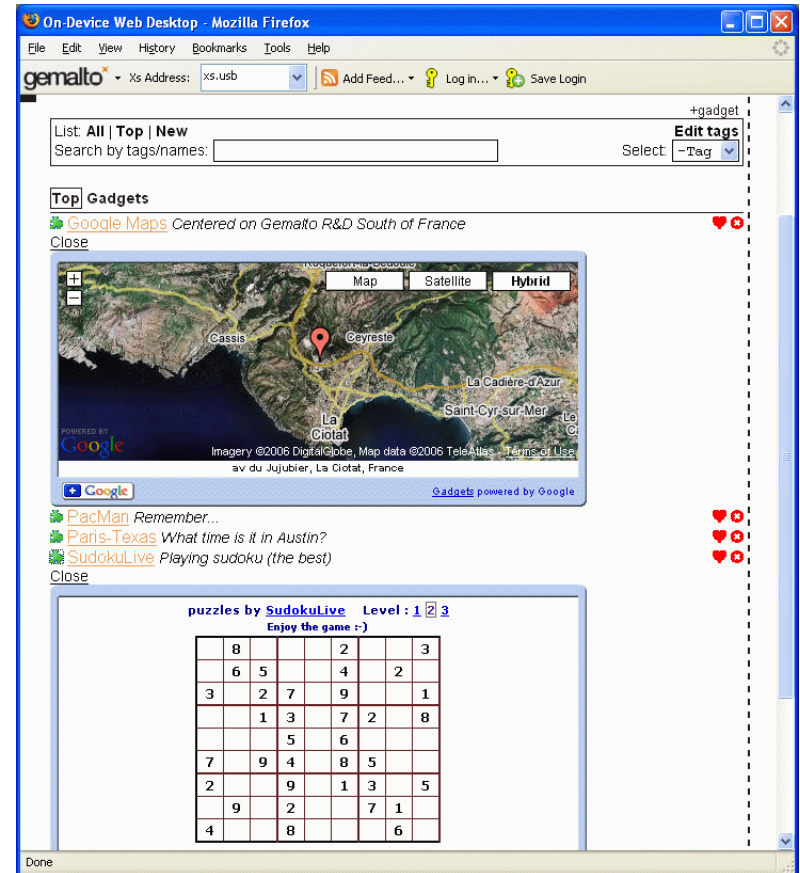
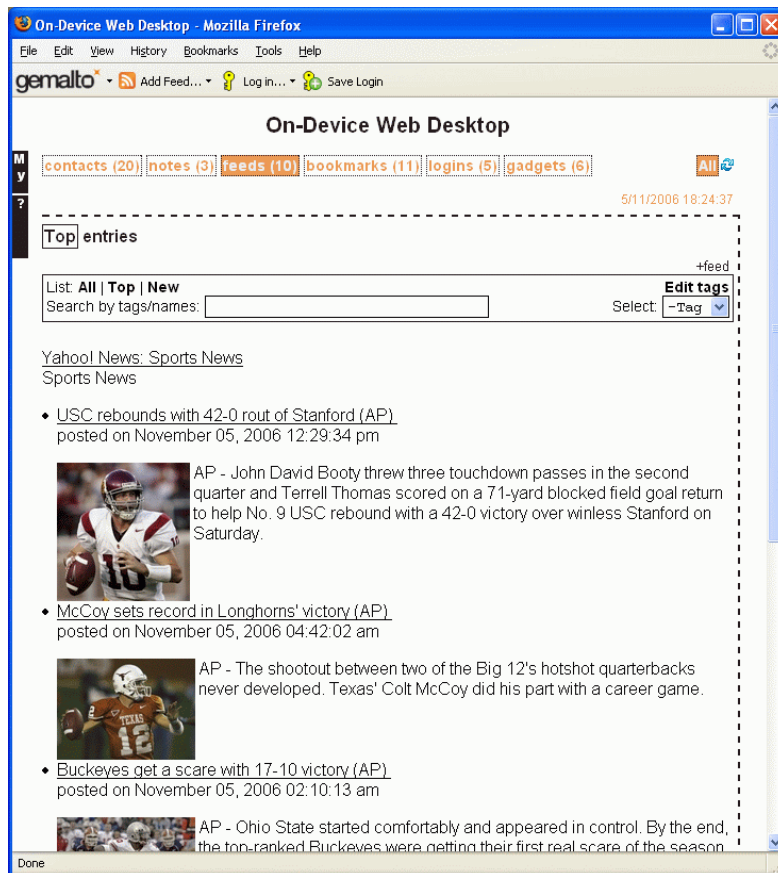
The demo is a prototype anticipating some NG Java Card technology-based functionalities

NG Java Card Technology-Based Web PIM

Mix personal data with online content

RSS Feeds

Google Gadgets



The demo is a prototype anticipating some NG Java Card technology-based functionalities



DEMO

A Next Generation Java Card Technology-Based Web PIM
The Web PIM application is called EWD (Embedded Web Desktop)

The demo is a prototype anticipating some NG Java Card technology-based functionalities

Agenda

Next-Generation Java Card Platform







NG Java Card Technology and Web 2.0

Demo: A NG Java Card technology-based
Web 2.0 PIM

Under the hood of the demo

Conclusion and perspectives

Responsibilities

	<p style="text-align: center;">Browser</p> 		
<p>Remote web servers</p>	 <p style="text-align: center;">Toolbar/extension Web page JavaScript</p>		<p>NG Java Card device with EWD Web app</p>
 <p>Plugins</p> 	<ul style="list-style-type: none"> ✓ Quick link to PIM ✓ Shortcuts to add personal data 	<ul style="list-style-type: none"> ✓ Queries EWD to insert / edit personal data in HTML pages ✓ Mash-es-up EWD provided personal data <i>and</i> Remote web server provided public data 	<ul style="list-style-type: none"> ✓ TCP/IP Web server ✓ Stores personal data ✓ Serves EWD application files: JS CSS, HTML, images ✓ HTTP API to get, add, delete personal data

The demo is a prototype anticipating some NG Java Card technology-based functionalities

In-browser Mashup: Overcome Same-origin Policy

- Principles
 - A single web page: EWD → From EWD
 - Information from another server: RSS → From Yahoo! News
- Issue: Same origin policy
- Solution: On-demand JavaScript programming language
 - Instead of issuing an XMLHttpRequest, insert a script tag in the HTML page (no same origin policy)
 - The script URL points to a online service, and contains the URL of the requested information
 - The obtained script contains the “write” directives to insert the feed content into the page



```
<script src=
"http://itde.vccs.edu/rss2js/feed2js.php?src=http
%3A%2F%2Frss.news.yahoo.com%2Frss%2Fsports&chan=y
&num=0&desc=1&date=y&targ=y&html=y">
```

```
document.write('<div class="rss_box">');
document.write('<p class="rss_title"><a
class="rss_title"
href="http://news.yahoo.com/i/755"
target="_self">Yahoo! News: Sports News</a><br
/><span class="rss_item">Sports
News</span></p>');
document.write('<ul class="rss_items">');
document.write('<li class="rss_item"> ...
```

Room for Improvement

- This demo is a **proof-of-concept**
 - Illustration of the guidelines: design 3-Tier, use AJAX and in-browser mashup
 - Illustration of interesting functionalities: rich user experience, rich content, offline and online mix, privacy and security
- Some additional functionalities to experiment
 - Login/password management and secure login
 - Embedding existing online applications to get same user-experience offline and online
 - Synchronizing embedded and online content
 - Smarter solutions to in-browser mashup of content
 - **List not closed...**

The demo is a prototype anticipating some NG Java Card technology-based functionalities

Agenda

Next-Generation Java Card Platform

NG Java Card Technology and Web 2.0

Demo: A NG Java Card
technology-based Web 2.0 PIM

Under the hood of the demo

Conclusion and perspectives

Summary: A New Platform

What you've learned from this session

- Next-Generation Java Card technology is **a full-fledged Java platform for embedded Web applications**
- Major evolution of the Java Card platform with a Java Servlet engine for dynamic Web application
- Ready for devices with standard high-level connectivity: USB, TCP/IP, HTTP(S)

Summary: New Web 2.0 Opportunities

What you've learned from this session

- NG Java Card Web 2.0 Applications
 - Alternative to **personalized applications** on remote servers
 - **Privacy of personal data:** personal secure storage, not transmitted to remote third-parties
- Additional benefits
 - Offline experience
 - NG Java Card Web applications can be as rich as RIAs

Summary: Guidelines for Efficiency

What you've learned from this session

- **Building rich Internet applications with a NG Java Card device** is possible
- 3-Tier architecture to limit the charge in the NG Java Card device
- AJAX development methodology to meet high-level user experience
- In-browser mashup of local and remote content to maintain privacy

Perspectives

New opportunities

- Hoping this talk has given you ideas of novel Web applications
 - With the Next-Generation Java Card platform
 - In the context of Web 2.0 and beyond...
- NG Java Card platform availability
 - Public Release of specifications scheduled by start 2008
 - Products to follow in 2008

For More Information

- **TS-5686: Next Generation Java Card Technology For Secure Mobile Applications**
 - Saqib Ahmad, Sun Microsystems, Inc.; Eric Vetillard, Trusted Labs; Florian Tournier, Sun Microsystems Inc.
- **BOF-5368: A Raconteur's Tour of Java Card Technology Development**
 - Seth Meltzer, U.S. Treasury/IRS; Doris Baker, dmb
- **Sun Booth: Gemalto's NG Java Card Technology demo**
- <http://java.sun.com/products/javacard/>



Q&A

Thank You!

Laurent Lagosanto
Jean-Jacques Vandewalle



gemalto
security to be free

JavaOne

Web 2.0 Applications on a Next-Generation Java Card™ Platform

Laurent Lagosanto
Jean-Jacques Vandewalle

Research Engineers
Gemalto

<http://www.gemalto.com>

TS-5203