Mashups and Modularity: Towards Secure and Reusable Web Applications

Antero Taivalsaari
Tommi Mikkonen

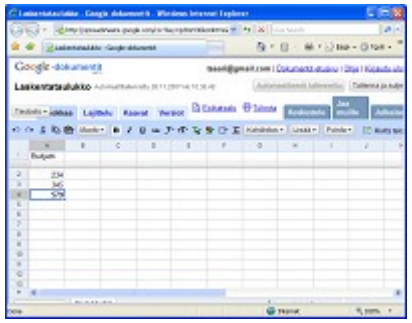
Sun Microsystems Laboratories
firstname.lastname@sun.com

http://research.sun.com/projects/lively

Evolution of the Web

1) Simple pages with text and static images only
(e.g., http://www.google.com)

2) Animated pages with plug-ins
(e.g., http://www.cadillac.com)



3) Rich Internet Applications
(e.g., docs.google.com)

What's Next?

Web Applications – Implications

- Web-based software will dramatically change the way people develop, deploy and use software.
- No more installations!
  - Applications will simply run off the Web.
- No more upgrades!
  - Always run the latest application version.
- Instant worldwide deployment!
  - No middlemen or distributors needed.
- No CPU dependencies, OS dependencies, ...
  - The Web is the Platform.

Unfortunately...

- The web browser was not designed for running real applications.
  - It was designed in the early 1990s for viewing documents, forms and other page-structured artifacts – *not* applications.
  - Programming capabilities on the web were an afterthought, not something inherent in the design of the browser.
- Various Rich Internet Application (RIA) technologies have been introduced recently to retrofit application execution capabilities into the web browser.

Web Development vs. Conventional Software

The Impedance Mismatch

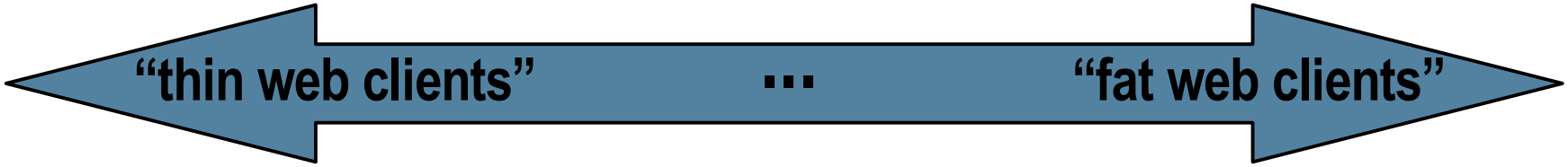
Web Development	Conventional SW Development
- Documents	- Applications
- Page / form oriented interaction	- Direct manipulation
- Managed graphics, static layout	- Directly drawn, dynamic graphics
- Instant worldwide deployment	- Conventional deployment
- Source code and text favored	- Binary representations favored
- Development based mostly on conventions and “folklore”	- Development based on established engineering principles
- Informal development practices	- More formal development
- Target environment not designed for applications	- Target environment specifically intended for applications
- Tool-driven development approach	- A wide variety of development approaches available

Landscape of RIA Technologies

Browser-based
- Ajax
- Google Web Toolkit
- Sun Labs Lively Kernel

Plugin-based
- Flash & Flex
- (Java FX, AIR)
- (Microsoft Silverlight)

Custom runtime
- Java, Java FX
- Adobe AIR
- Silverlight



- Run in a standard browser
- No plug-ins needed
- Platform-independent
- Browser-based UI

- Browser plug-in required
- Custom UI

- Custom execution engine required
- Runs outside the browser
- Custom/native UI

Technologies in the web browser serve as the lowest common denominator!

Beyond Google Docs and Other Desktop-Style Web Applications...

Software as a Social Mashup

Web 2.0 – What Is It Really About?

- *Interaction*. Bringing back some of the best qualities that desktop software had before the Web, such as direct manipulation, instant feedback, piecemeal display updates.
- *Collaboration*. Allowing the users to work with each other in a “social” fashion, and share the same data, applications and services over the Web, regardless of their physical location.
- *Mashups*. Being able to combine content available on the Web in novel ways.

An Important Observation

- Web applications are **not** just conventional desktop applications running in the web browser.
  - Not necessarily just word processing applications, spreadsheets, e-mail or instant messaging clients, ...
- The Web enables the creation of entirely new types of applications and services that combine content from other web sites dynamically.
  - This would not have been possible with conventional shrink-wrapped applications distributed in binary form.

Mashups

- *Mashup*: A web site that combines content from more than one source (multiple web sites) into an integrated experience.
- Mashups leverage the power of the Web to support worldwide sharing of content that would not have been easily accessible or reusable before the Web.
- In principle, the content to be combined can be anything (text, source code, maps, video, blogs, product reviews, price data, ...) as long as it can be meaningfully combined with other content.

Examples of Mashups

- Chicago Police Department crime statistics mashup (http://chicago.everyblock.com/crime/)
- Parking availability mashups (e.g., http://www.parkingcarma.com/)
- Traffic tracking and congestion mashups (e.g., http://dartmaps.mackers.com/)
- Real estate sales and rental mashups (e.g., http://www.housingmaps.com/)

Observations on Mashups

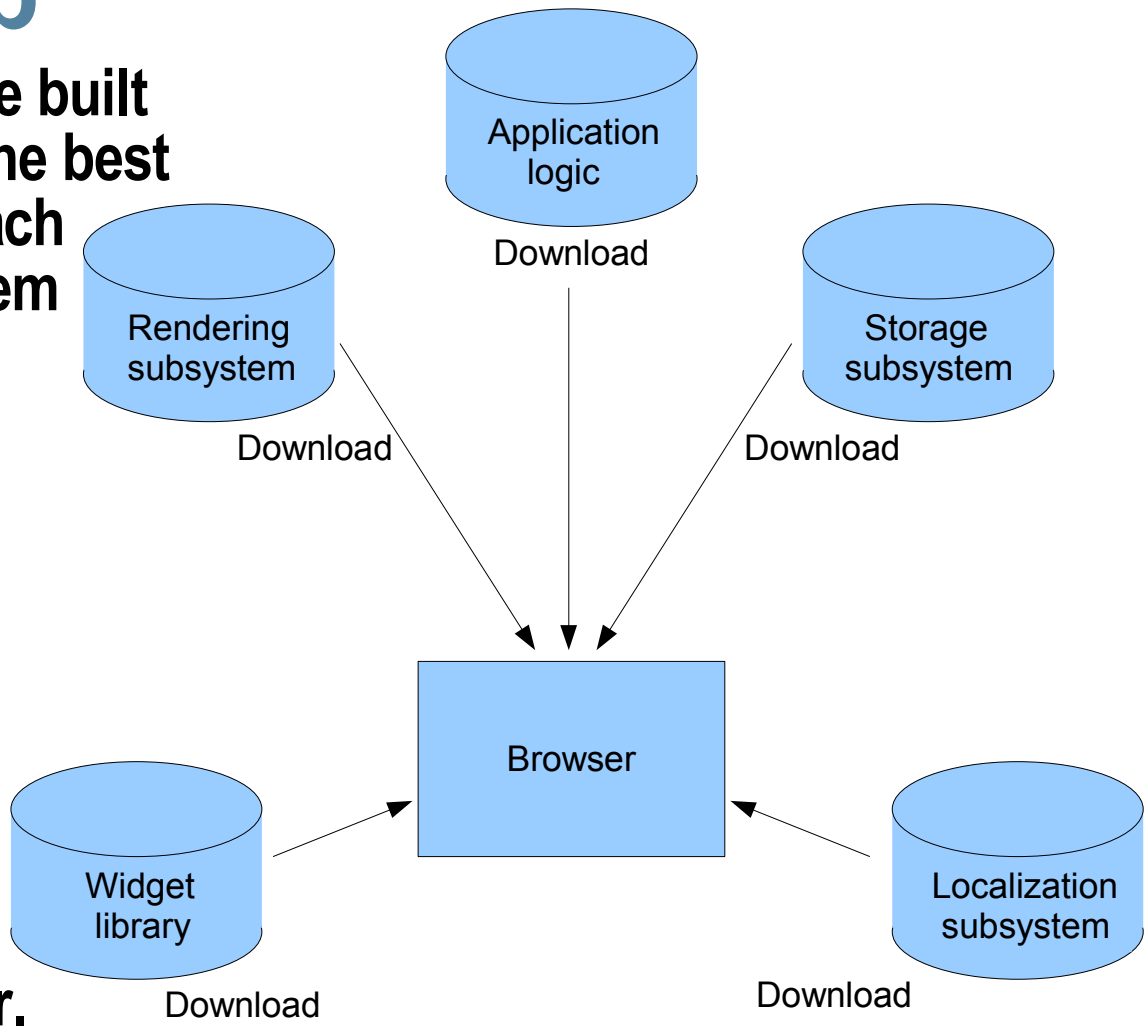
- Today, most mashups are built around *maps*.
- However, in principle the content can be anything as long as it can be digitalized and shared over the Web.
- Mashups are usually generated dynamically with no static linking; textual representations such as HTML, XML or JSON favored.
- In principle, it would be possible to build software as a mashup as well.

The Future of Software as a "Social Mashup"

In the future, software will be built by dynamically combining the best available components for each purpose by downloading them dynamically from different web sites.

No static linking; everything downloaded on demand.

Software development will be an inherently social activity between developers who do not necessarily know each other.



Today's web browsers do not support these kinds of applications yet!

Mashups vs. Modularity

- Modularity is an important *social* aspect of a system.
  - Modularity is the enabling factor for large-scale software development and reuse.
  - If there is only one developer, there would be little or no need for modularity.
  - Modularity makes it possible for a large number of developers to work without interfering with each other.
  - Without well-defined interfaces, implementation changes would be visible to everybody.
- The lack of proper modularity is a key deficiency (along with security issues) that prevents web application and mashup development in a truly social fashion.

Software As a Mashup: Two Main Problems

- *The lack of modularity / well-defined interfaces.*
  - Prevents developers from easily sharing and reusing code developed by others.
- *The lack of a flexible security architecture*.
  - Prevents developers from easily and securely downloading and combining source code from multiple web sites across the planet.
  - The *Same Origin Policy* restricts access to other web sites, see http://en.wikipedia.org/wiki/Same_origin_policy

Additional Problem Areas

- During our project, we have discovered problems in various areas related to the use of the web browser as an application platform:
  1) Usability and user interface issues
  2) Networking and security issues
  3) Browser interoperability and compatibility issues
  4) Development style and testing issues
  5) Deployment issues
  6) Performance issues
  7) Software engineering issues

Additional Problem Areas

- For details & possible solutions, read our papers:
  - "Web Applications – Spaghetti Code for the 21st Century"
    http://research.sun.com/techrep/2007/abstract-166.html
    (presented in the SERA Conference, Prague, Czech Republic, August 21, 2008)
  - "Web Browser as an Application Platform: The Lively Kernel Experience"
    http://research.sun.com/techrep/2008/abstract-175.html
    (presented in the SEAA Conference, Parma, Italy, September 4, 2008)

Related Work

- IBM SMash (De Keukelaere, Bhola, Steiner, et al.)
- Microsoft MashupOS (Wang, Fan, Howell, et al.)
- Google Caja (Miller, Samuel, Laurie, et al.)
- Various Interface Description Languages (IDLs)
- Related web application and RIA development environments (Adobe AIR, Java FX, Microsoft Silverlight, ...)

Mashup Development Tools

- Dapper, http://www.dapper.net/
- Google Mashup Editor, http://code.google.com/gme/
- IBM Mashup Center, http://www-01.ibm.com/software/info/mashup-center/
- IBM Project Zero, http://www.projectzero.org/
- Intel Mash Maker, http://mashmaker.intel.com/
- LiquidApps, http://www.liquidappsworld.com/
- Microsoft Popfly, http://www.popfly.com/
- Mozilla Ubiquity, https://wiki.mozilla.org/Labs/Ubiquity
- Open Mashups Studio, http://www.open-mashups.org/
- Yahoo Pipes, http://pipes.yahoo.com/

Academic & Less Widely Known Tools

- d.mix (Stanford University), http://hci.stanford.edu/mashups/
- Marmite (Carnagie Mellon University), http://www.cs.cmu.edu/~jasonh/projects/marmite/

  ========
- Anthracite, http://metafy.com/products/anthracite/
- C3W (Clipping, Connecting and Cloning for the Web), http://www.iw3c2.org/WWW2004/docs/2p444.pdf
- Internet Scrapbook, http://www.sigchi.org/chi97/proceedings/short-talk/as.htr
- And many others...

Conclusions

- Like it or not, the Web is increasingly the platform of choice for advanced software applications.
- Web-based applications have major benefits: no installation or upgrades needed, instant worldwide deployment without middlemen.
- Web-based applications will dramatically change the way people develop, deploy and use software -> paradigm shift!
- Web technologies need to evolve in a direction that allows software to be created by dynamically combining the best available components from all over the world -> software as a “social mashup.”

Thank You! Questions?

Tommi Mikkonen
Antero Taivalsaari