A look at the concept of an “app”, as seen from the point of view of the web and its standardised technologies.

Something you really **must** have, right?

- Buzz on town the latest years …

Apps are the "talk of the town". "What apps do you have?" "See the new app I found!"

What is this all about, really

- What are our options?
- What are the consequences of making a choice?
- Can we wait and see?
- Or do the jump into this novelty?
- Is there some alternative?

This talk gives a look at:

- What do we mean: “apps” and “contents”
- “Apps” and “web apps”
- Technology and standards

How should we understand what kind of animal an app is? Is it something fundamentally new, technologically speaking? Or is it mainly a marketing concept – something that has created new income streams for certain players in the telecom/Internet sector?

Difference: Contents – apps

First stab: *dictionary* view

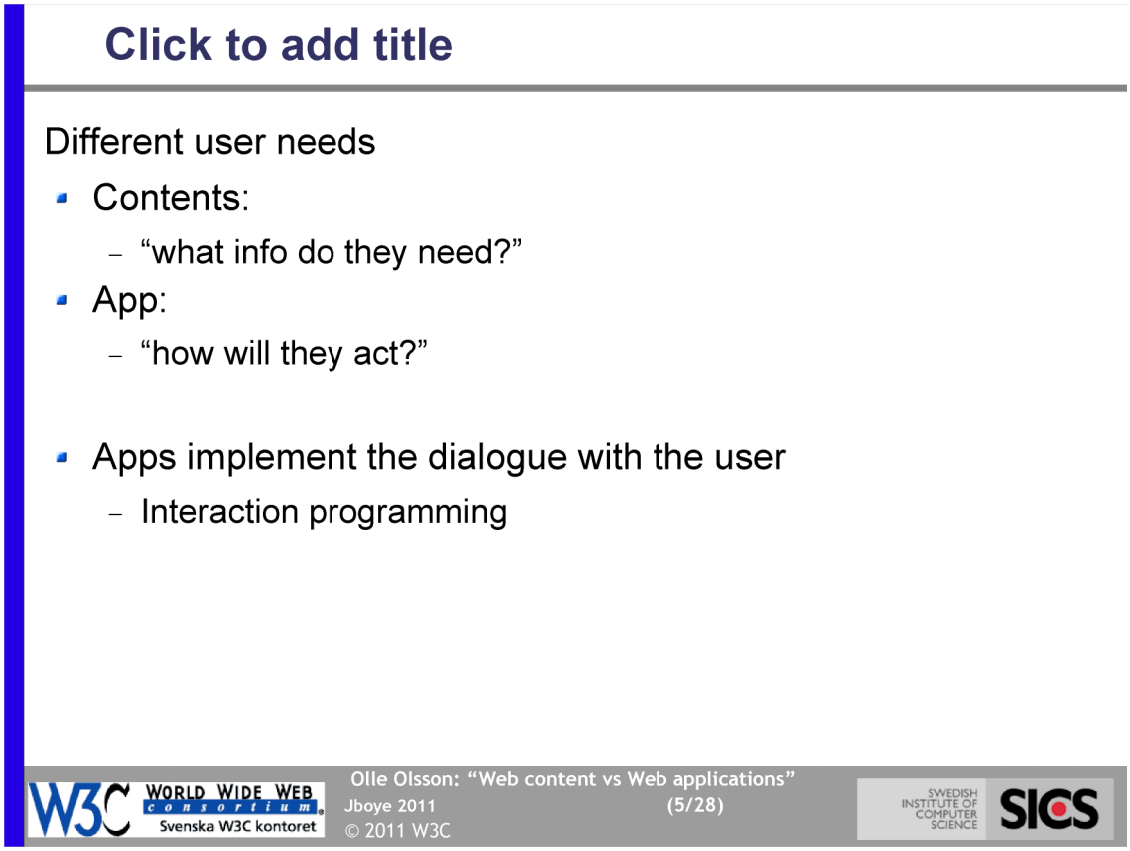
- Content
  - "information"
- Application
  - "processing"
  -

Second stab: *action* view

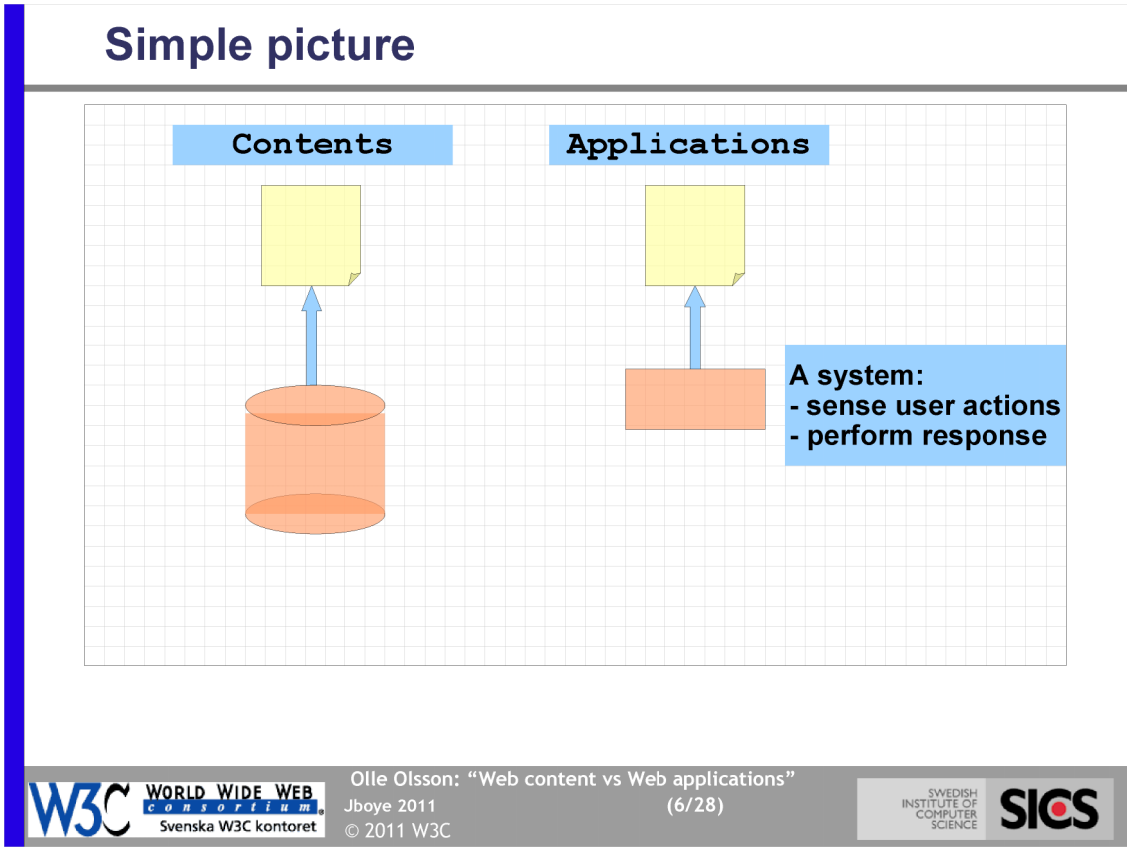
- Content
  - Identify user needs; structure/organize information; create information
- Application
  - Identify user needs; define architecture & design; do programming

What is in a name?

What does it mean for ME as a content/service provider?

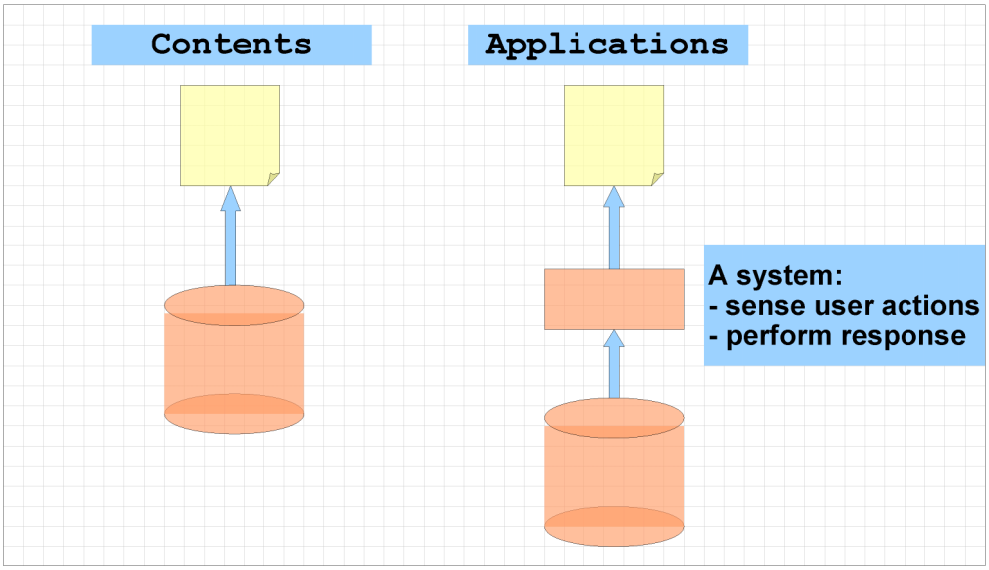


A stronger focus on user *actions*, compared to focus on what information is valuable/relevant for the user.

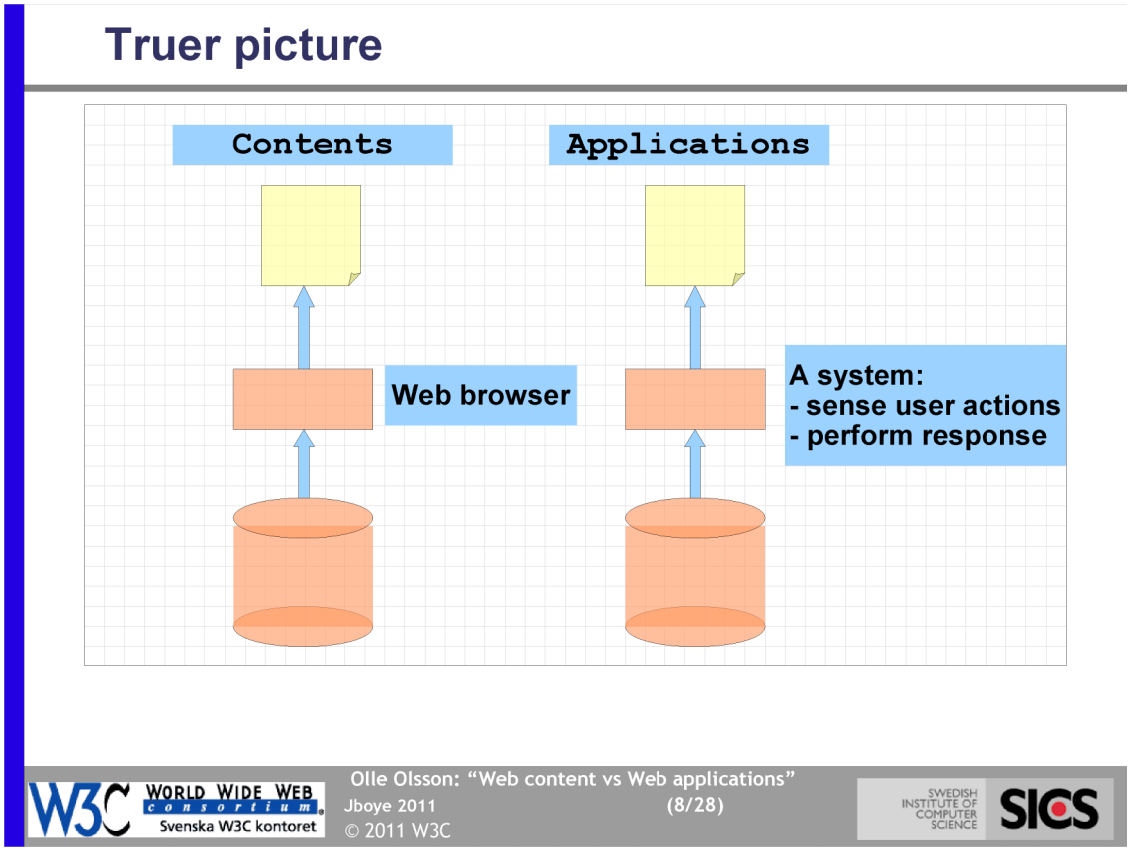


A simplistic graphical view of the "architecture" (in a very vanilla sense).

Better picture



Apps often fetch data across the net, so there is often a content repository in the backend, a repository that is used by the app.



The web browser can be seen as a "mega app".

But we do support user actions ...

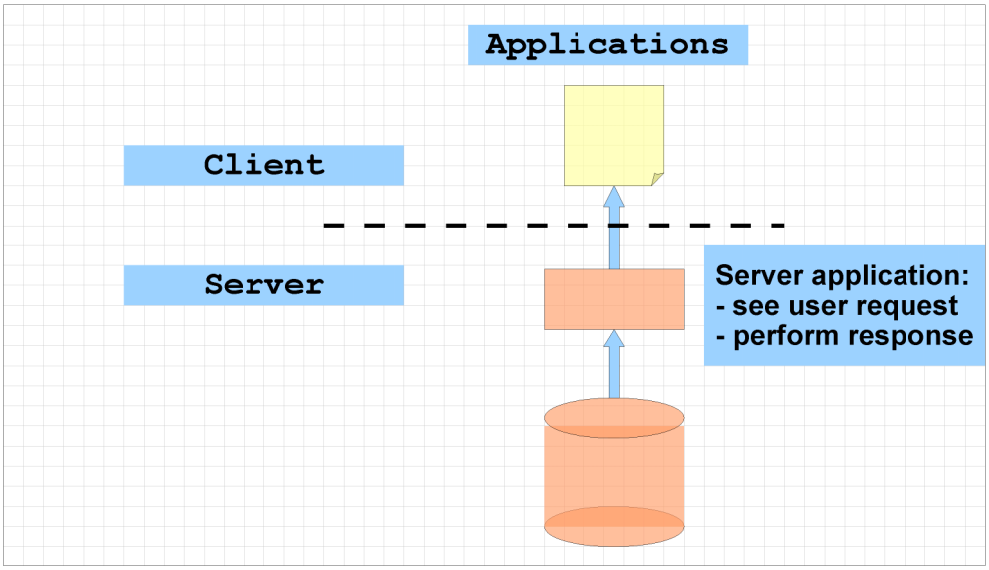
1) Web browser … big generic “app”
   - Handles user events
   - in a standard way
   - … dialogue handling
   - automatically available

2) Navigation structure
   - Defines (dialogue) state change in a local context
     - Links within a page
   - Defines a global space of (dialogue) state changes
     - All actual/potential links in all delivered contents

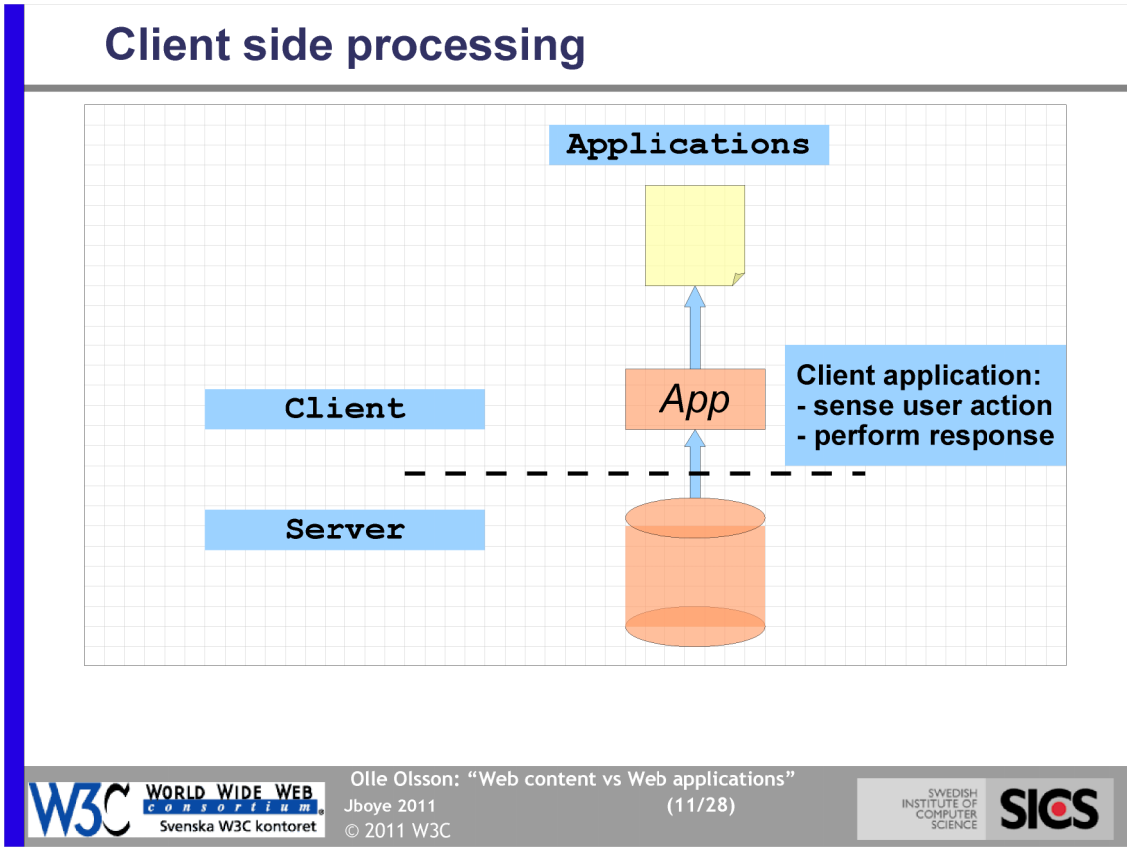
User actions supported:
- by the web browser
- through the navigation structures we create.

So we have always cared for user actions.

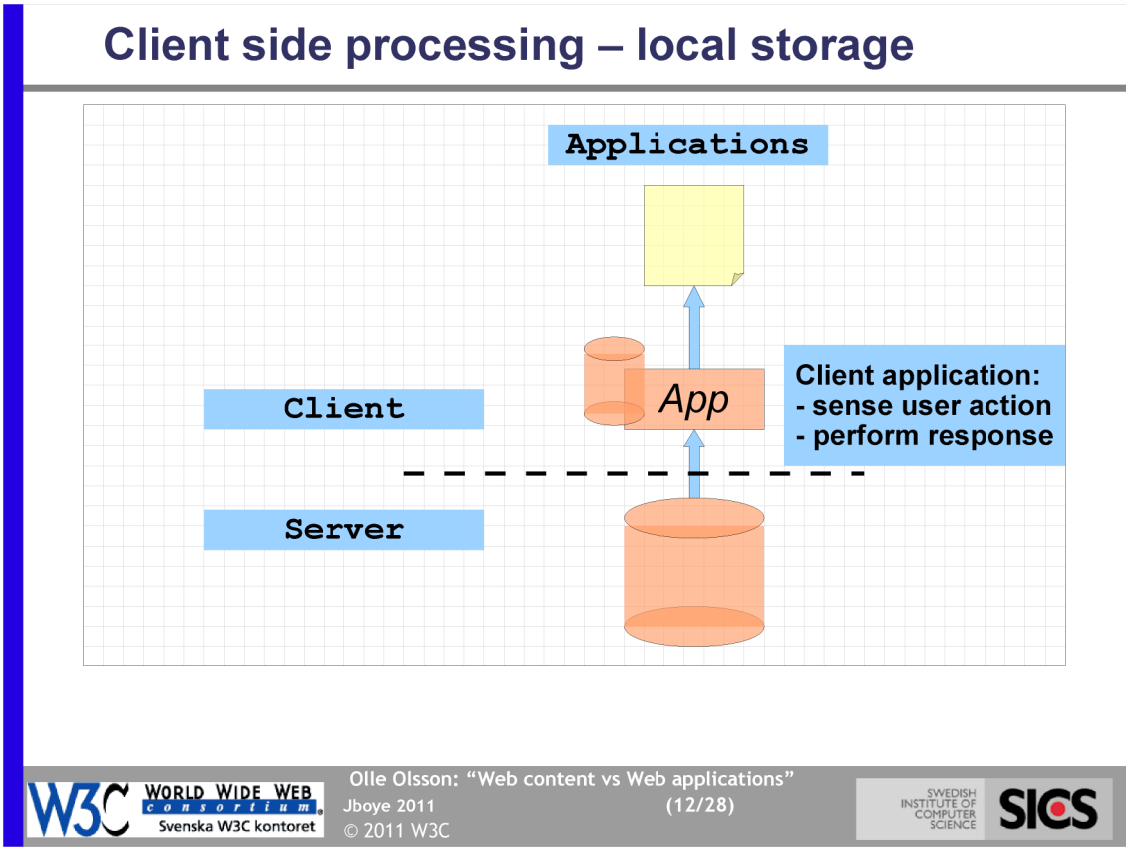
Server side processing



Server side actions (CGI-scripts, Servlets, …) are often context-dependent processing components. So we have some kind of "app behavior" supported at the back end.



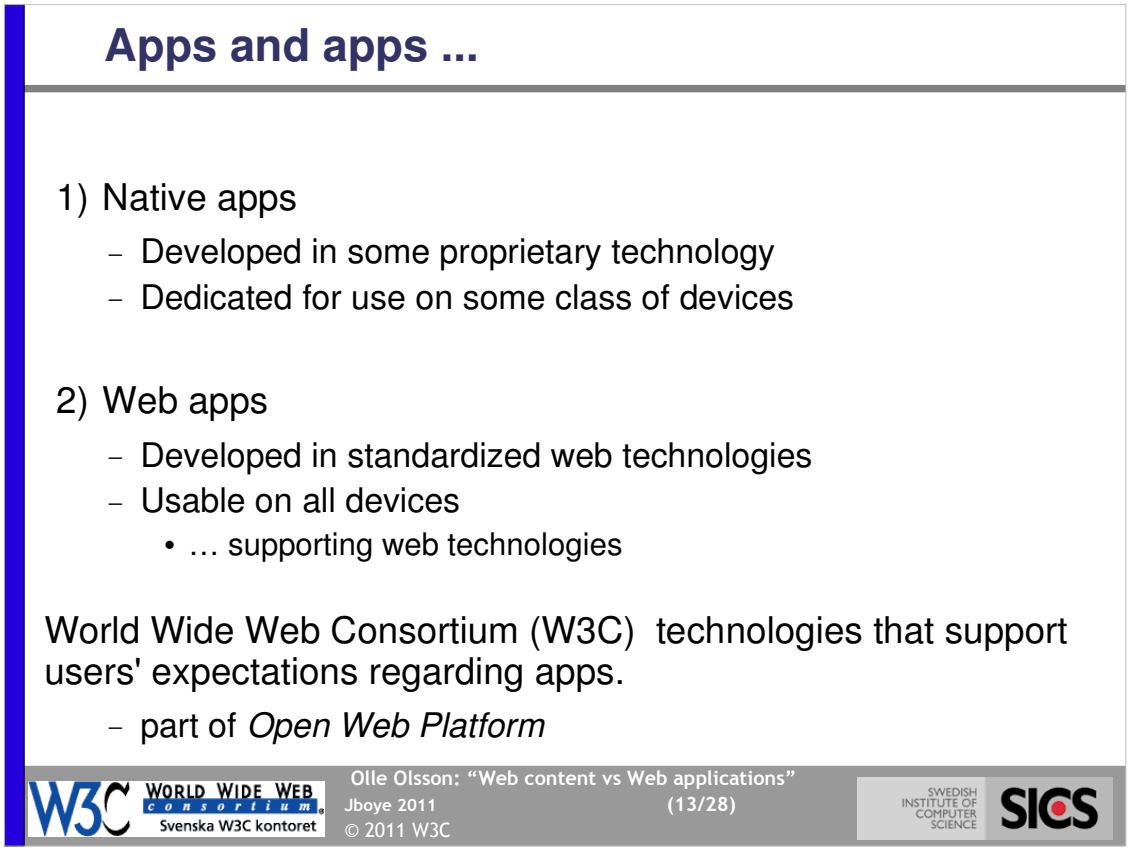
If this processing is moved from the backe end to the front end (from server to client) we have what is called an "app".



And this client side app can have a (client side) local content storage that it accesses to speed up response time.

Now, such approaches can actually be supported by the web technologies we have available today!

The popular apps can be seen as examples of requirements for a full-fledged front-end app experience. And these requirements are part of the web technologies tool-kit



The early hype concerned "native apps" that were not portable across platforms.

If we use web technologies for such apps, then we get portability, and this we can call "web apps".

World Wide Web Consortium

- Develops and governs web standards
- Non-profit Consortium, involving industry, academic, and public participation
- Mission: Leading the Web to its full potential
- Creating royalty free standards for the World Wide Web since 1994
- Many activities, check http://www.w3.org/

"To standardize and foster the deployment of an universal open platform for data, documents and applications on the Web that is suitable for human to machine, machine to machine and, ultimately, for human to human interaction."

W3C is driven by the needs of different stakeholders in the web.

Standardisation work in W3C is performed by persons from the W3C members.

Members are major vendors (Microsoft, HP, IBM, Google, Mozilla, Opera, SAP, etc), as well as organisations who are users of web technology (Boeing, Chevron, …)

As companies put effort into work at W3C only if they see a value and a need, the areas targeted by standardisation are vital for the continued growth and value of the web.

Standardisation by consensus, meaning that all participants in a standardisation of a specific technology has to agree completely on what the standard says.

Open Web Platform

- More than plain old HTML, CSS, HTTP
- Platform for applications, support for P2P, …
- Key component: HTML5
  - *HTML5 in narrow sense*: HTML as an encapsulating framework for an open set of specialised extensions
  - *"HTML5" in a broader sense*: the "narrow sense" HTML, ***plus*** all extensions that are on development plan.

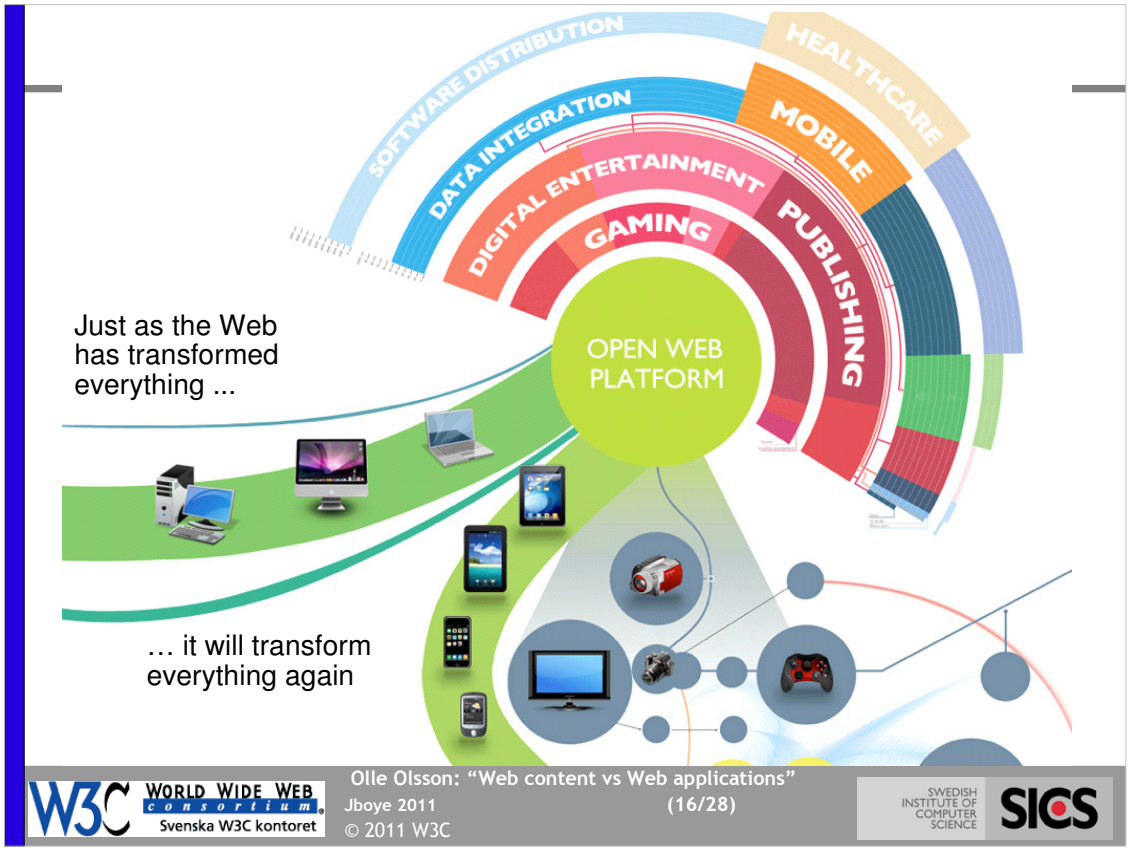
    … effectively the "Open Web Platform"
  - *HTML5 Recommendation* planned for 2014

HTML5 is much talked about now. It will provide a frameworks for strong, flexible and efficient content and service delivery.

The HTML Working group developing HTML5 has strong participation from all browser vendors. They provide their experience from implementing web technologies, so the HTML5 proposal is well grounded in practical experience. They have deep insight in what their customers ask for in terms of functionality, so the Working Group has access to all critical needs. And the vendors implement the draft proposals of the standard, before the standard is officially accepted.

This means that the modern browsers you use today already implement most of those parts of the HTML5 specification that are regarded as "stable".

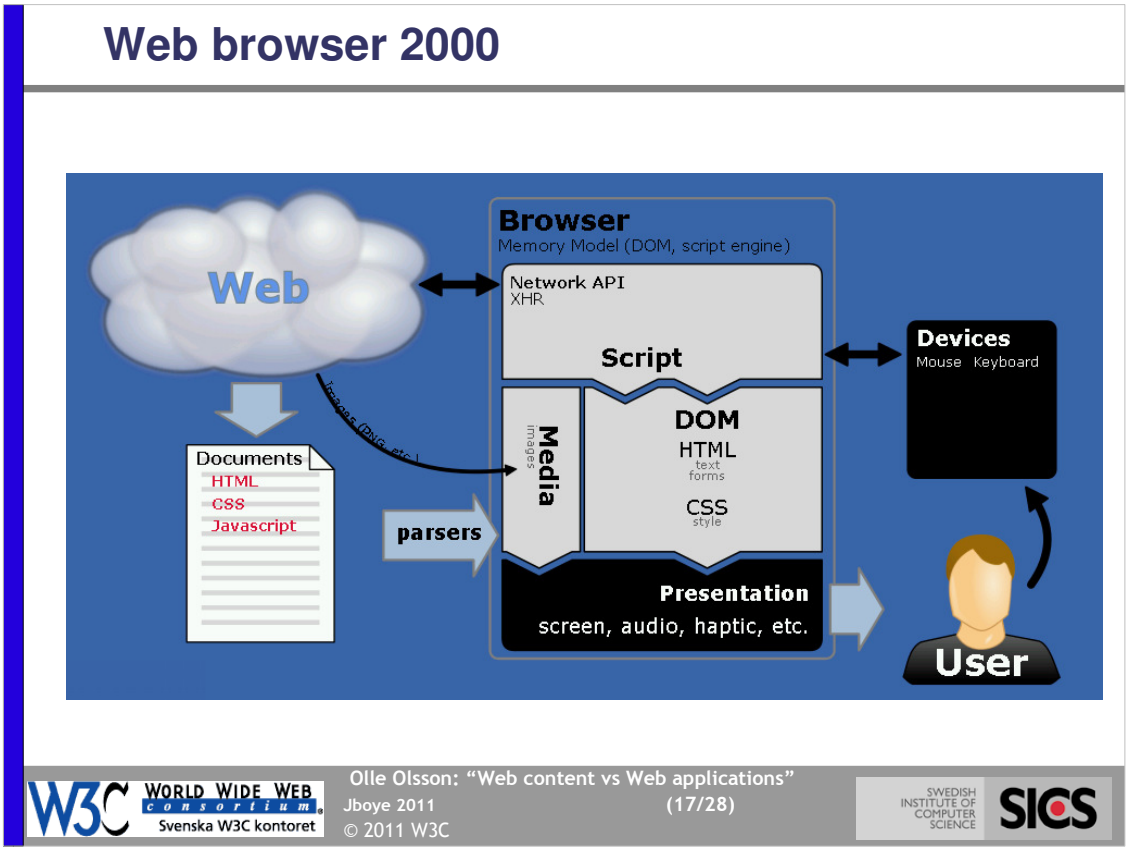
Why not experiment and explore HTML5 right now! Find examples on the web, and see its look-and-feel.



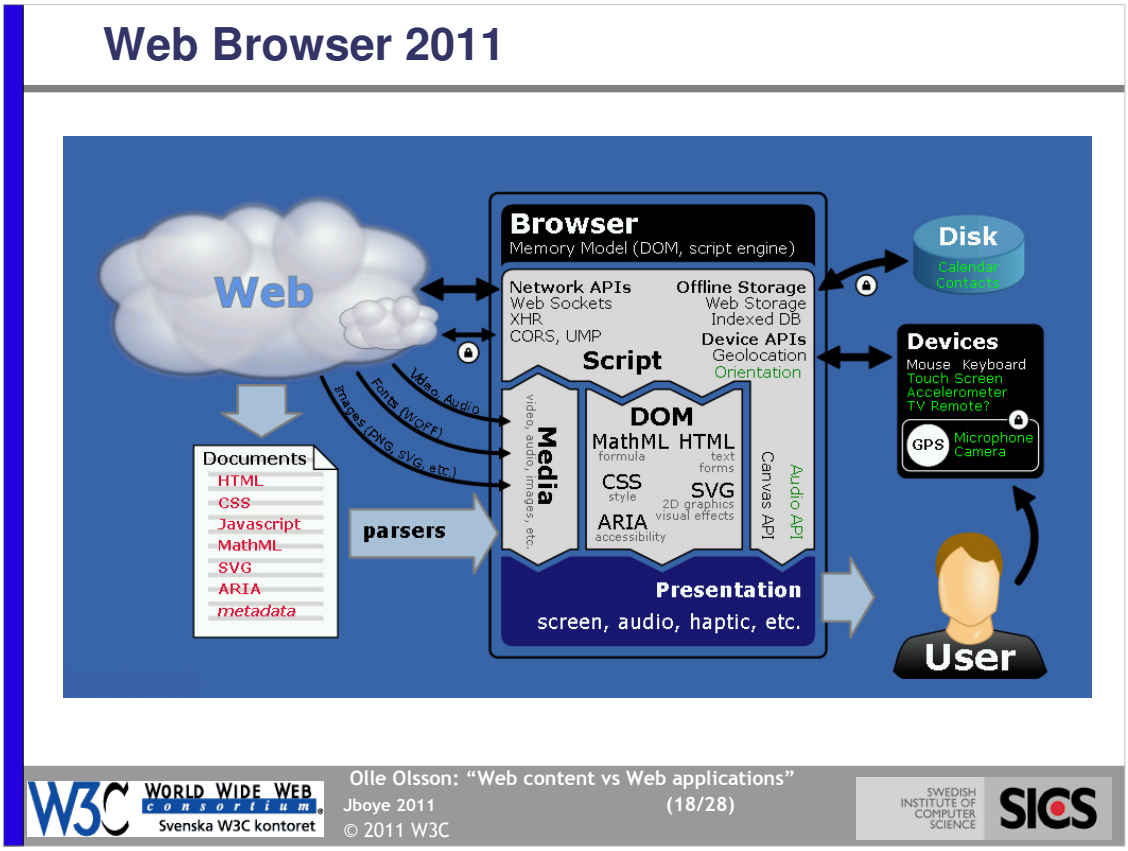
Not only for classical devices/hardware platforms.

Also for Phones.

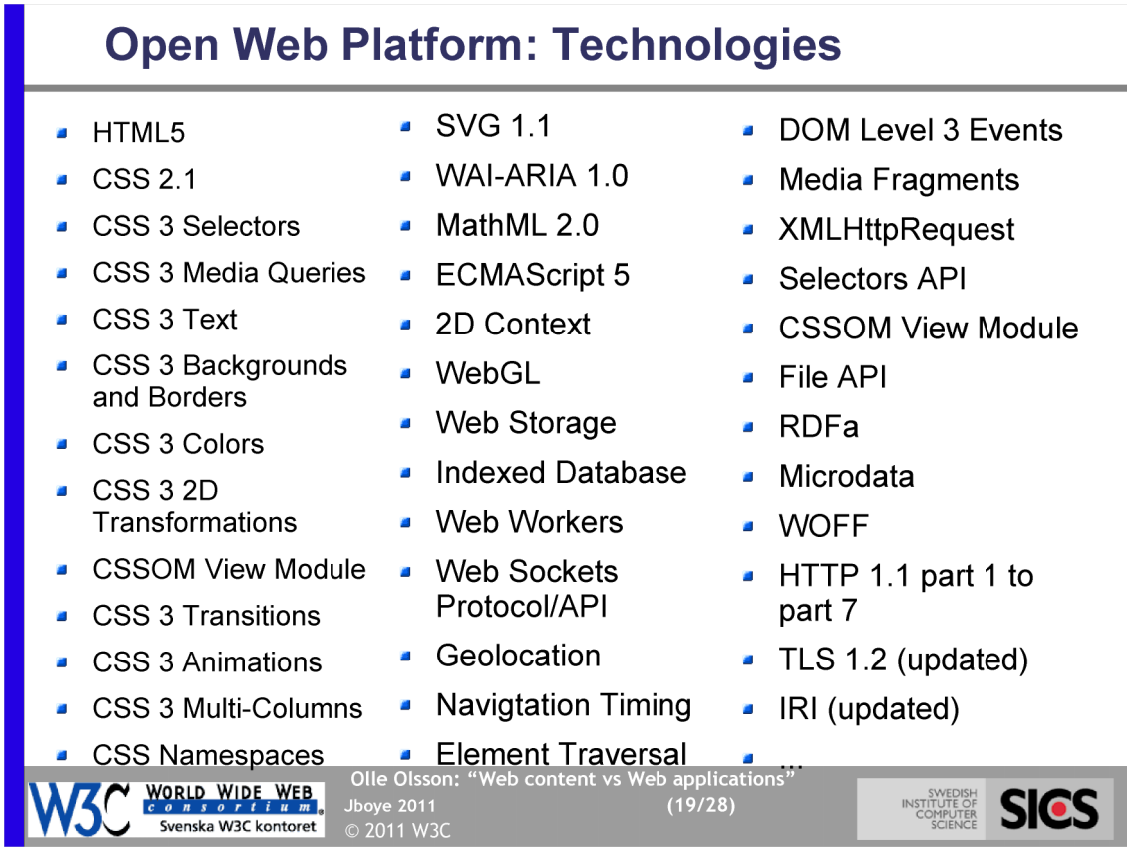
And for emerging new devices: TV, game consoles, Internet devices in cars, Internet of things, ...



The simple web browser ten years ago.



… has much more inside nowadays. Support for a multitude of special technologies. Small technologies that are standardised, or in the process of being standardised.

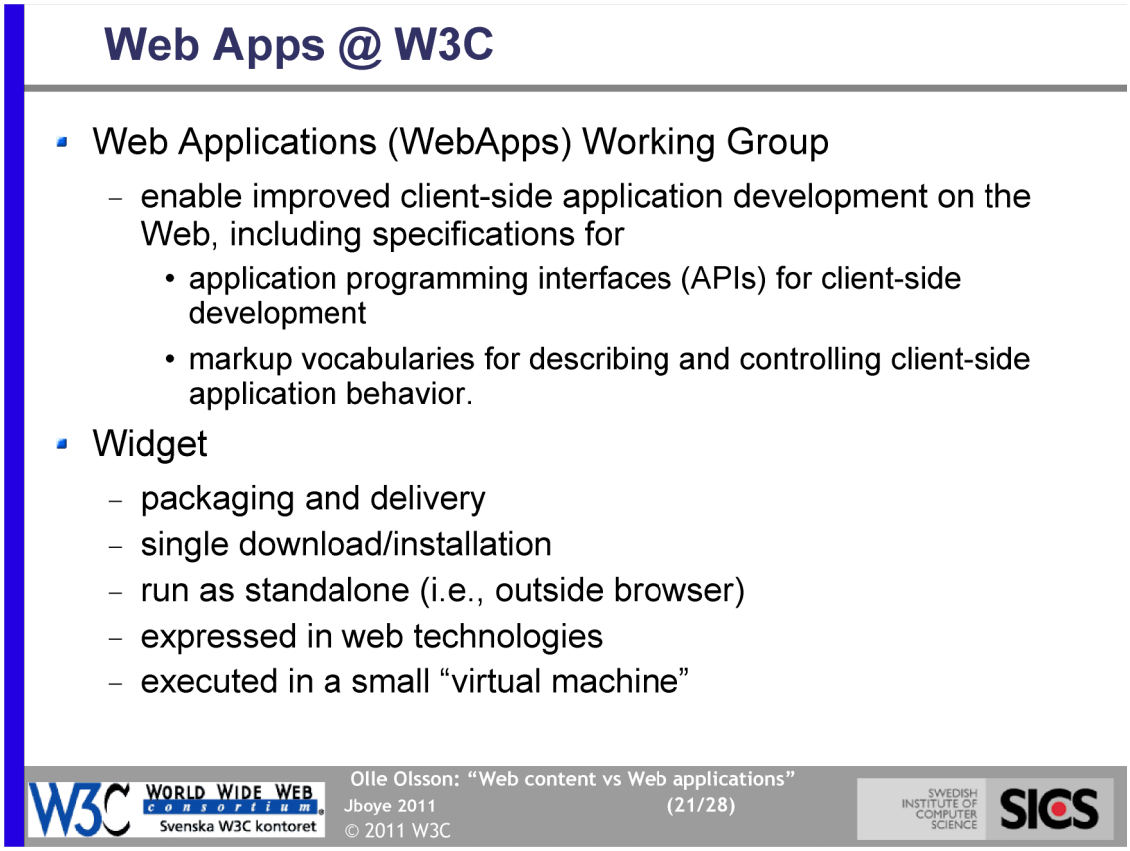


A few of the technologies that can be used in moderns web-based content and service delivery.

Multifunctional platform

- Support for content
  - Delivery, tailored presentation
  - Web of documents
  - Web of data
- Support for interaction
  - Tailored interaction; applications (and contents)
  - AJAX
- Applications:
  - Hosted in web browser
  - Hosted on device

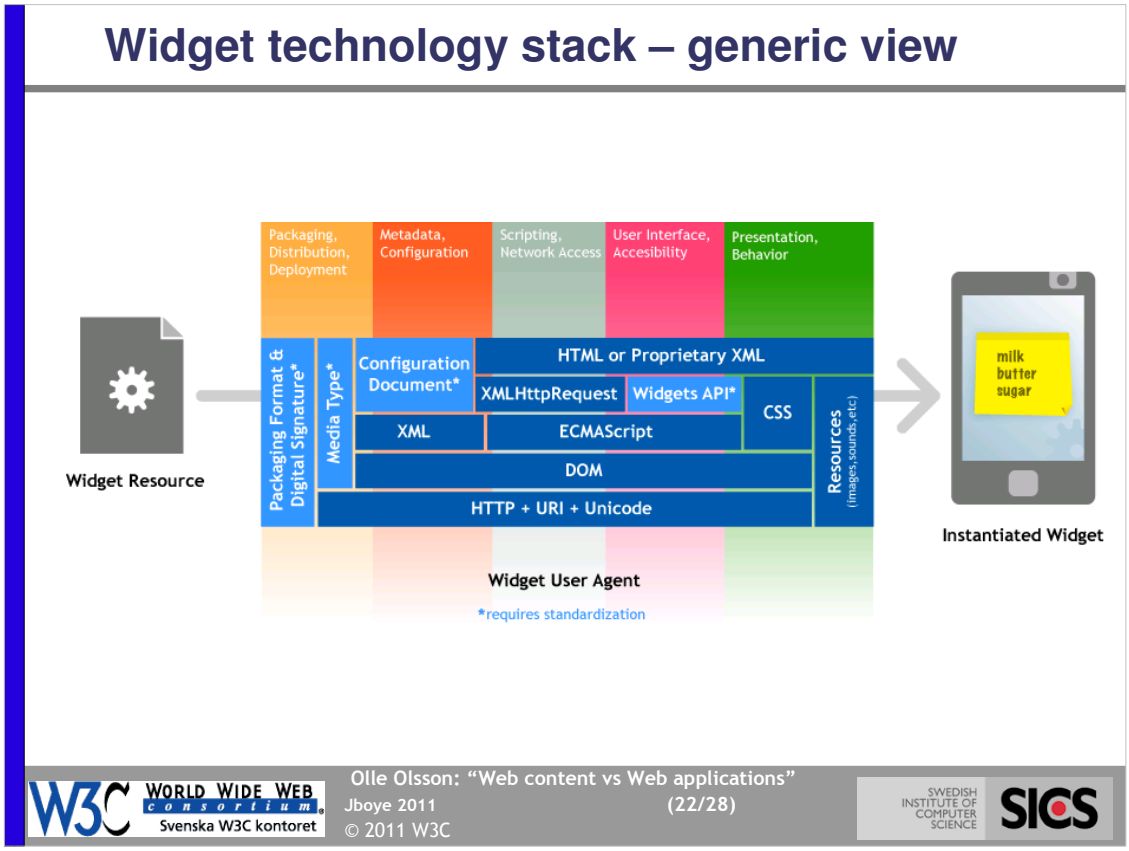
All those technologies, so we can support many new needs, and new ways of creating an infrastructure for content and service delivery.



Web apps as a technology sector addressed at W3C.

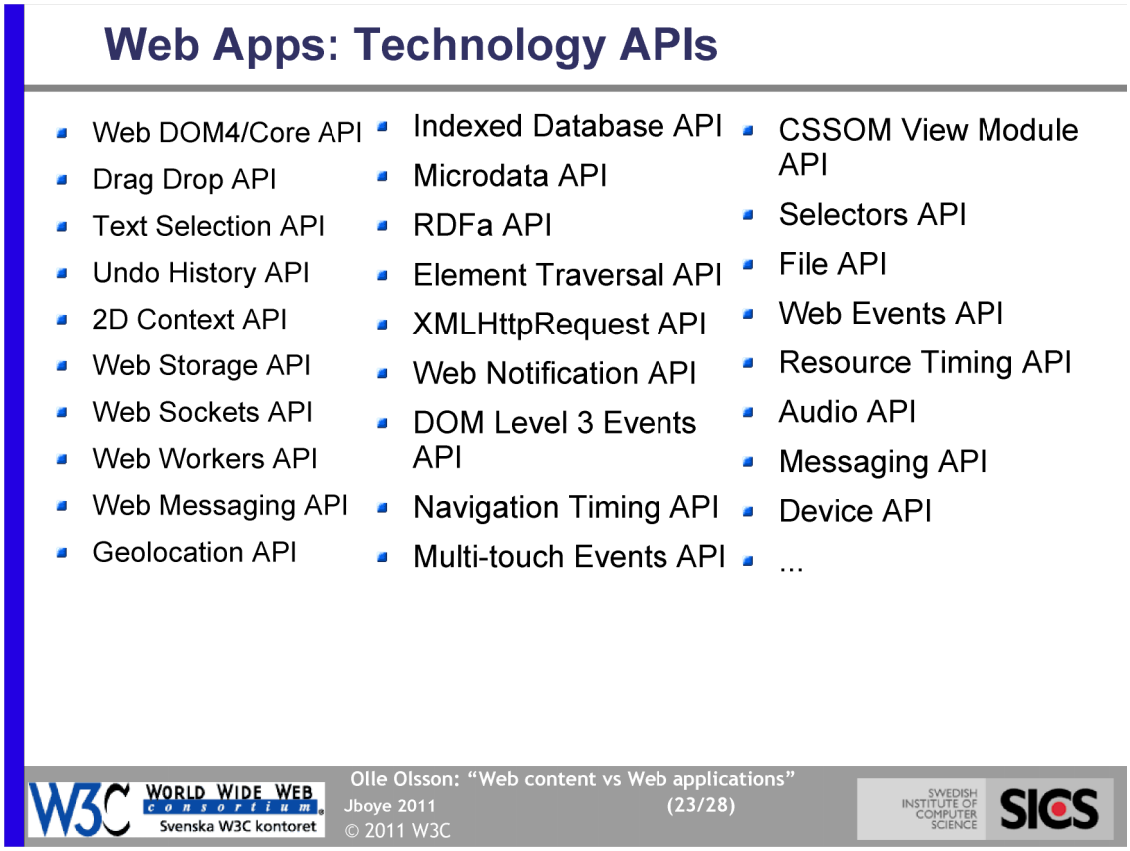
Basic idea: re-use technologies that have been standardised and used on the web … what we definitely can call "proven technologies".

Some special needs for Web Apps, as they are stand-.alone containers of functionality – do not have to be run in a full-scale browser (though they can run there also).



A view of what the "virtual engine" for Web Apps can look like.

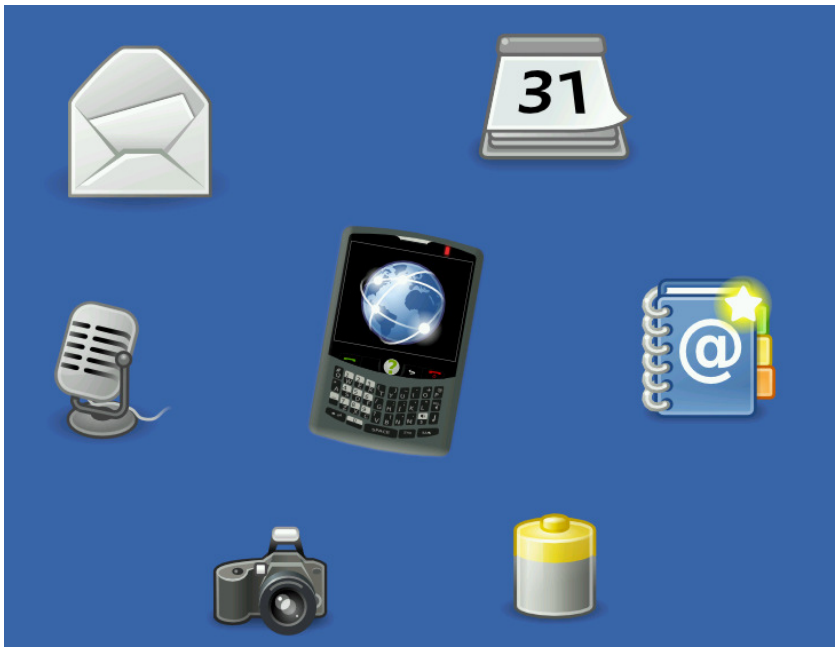
Mostly off-the-shelf re-use of client side engines for various standard technologis (like CSS). So implementations of these exist, for instance in all ordinary web browsers. By re-combinaing these components, and excluding the "chrome" of the web browser, we get a client side engine for running Web Apps.



Some of the technologies that are relevant for Web Apps.

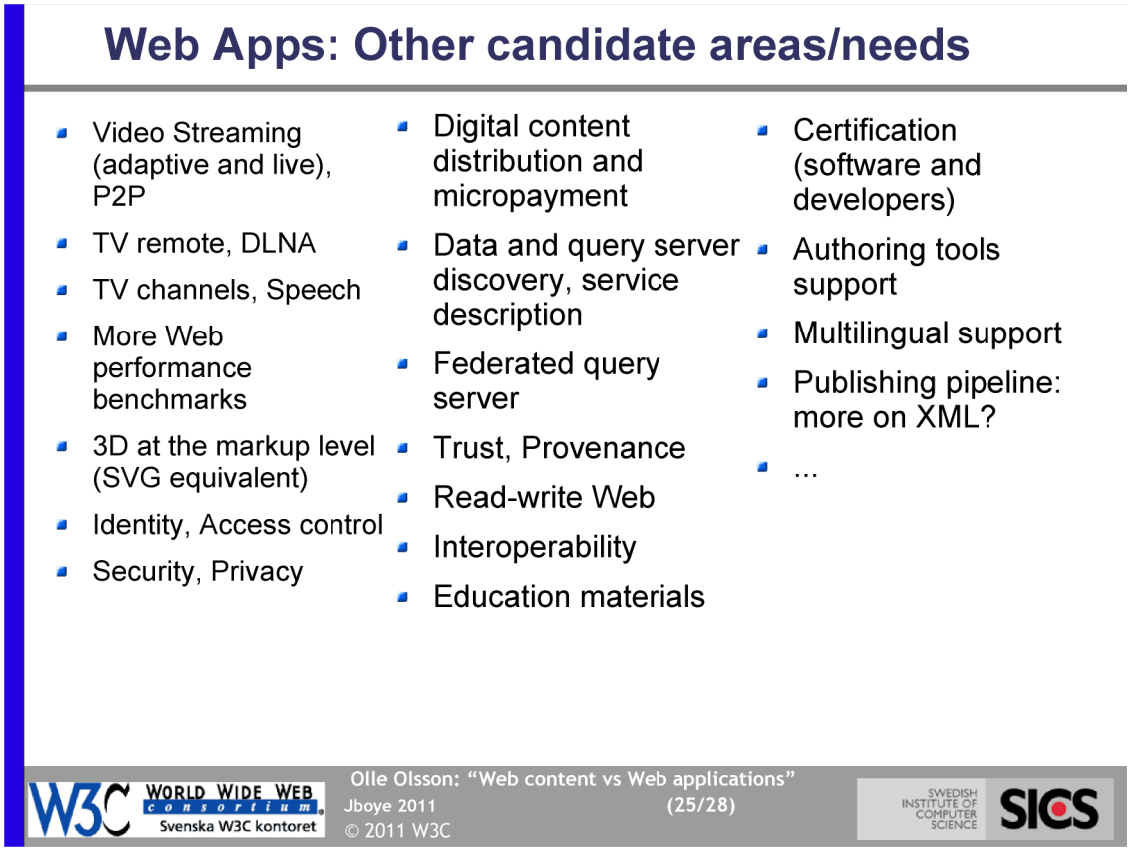
Most have been created to support applications embedded in web contents (think: "web 2.0"), but now they can be re-used for stand-alone web apps too.

Web Apps: Device APIs



Can access functionality implemented in a phone, functionality that we only need a standardised API to.

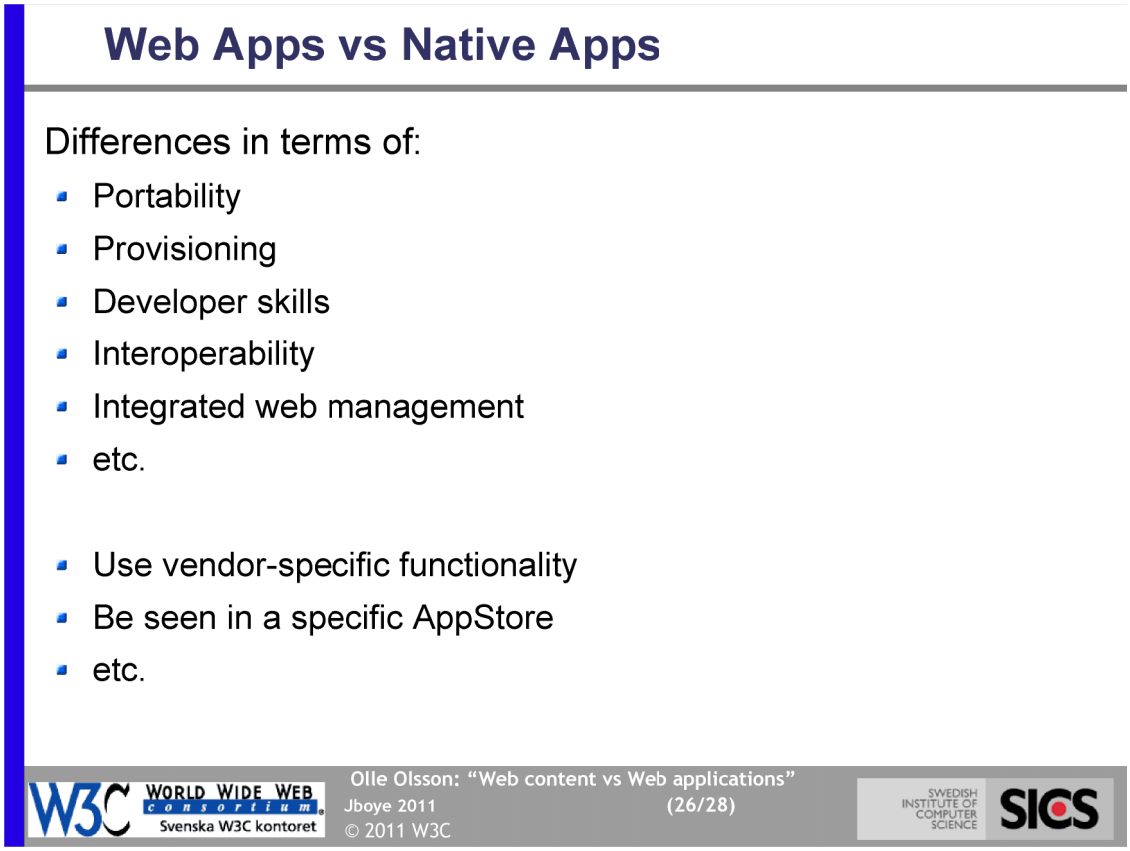
With such API:s we can create new added-value interfaces to such basic functionality … for instance creating a virtual address book, by combining the address book data in the phone's own address book, with address data from some web resource!



Doing a reuirements inventory has identified a large number of areas that we need to resolve before the App technology achieves all that we need from it.

This is not only what should be done for the Web Apps! It is also a sign that the different technologies for native apps still lack functionality. E.g. how can one handle security in a general way, when native app technology either do not offer it, or it is shaped in some proprietary way that is not harmonized with the security model you use on your web site?

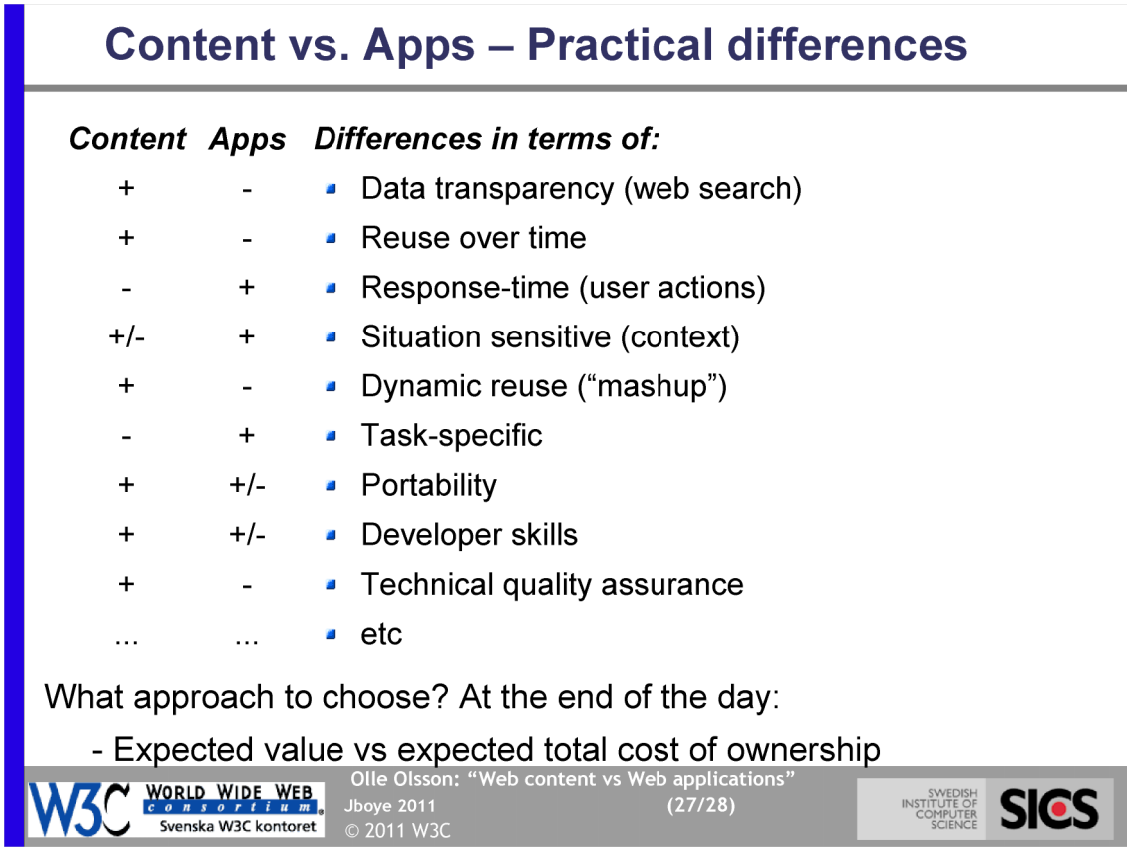
Web Apps have a potential to be more easily integrated with and interoperable with your ordinary web site and its web contents --- compared to native apps.



What are the advantages and disadvantages for native apps vs web apps?

Some obvious examples mentioned.

This is a topic that has been discussed quite a lot on the web, so do some web search if you want to find more pro/cons arguments.



And what are to be said about contents-centric solutions (the web as we are used to it) vs app-centric solutions?

Much can be said. Here I just mention a few obvious dimensions.

Click to add title

Thank you for your attention