Methods for developing with Sakai, serverside and datafeeds with fearless javascript

AuSakai 2008

Introduction

- Ian Boston
  CTO, CARET, University of Cambridge
  Backend developer
- Nicolaas Matthijs,
  CARET, University of Cambridge
  Frontend developer

The Plan

- [ ] Introduction
- [ ] Architecture
- [ ] Demos and examples
- [ ] Hands on: Back end
- [ ] Hands on: Front end

Introduction

- Who are you ?
- Sakai Development
- Drivers for Change

Sakai App Development

- Developer Driven
- Barriers to UI Developers

Stakeholder

Designer

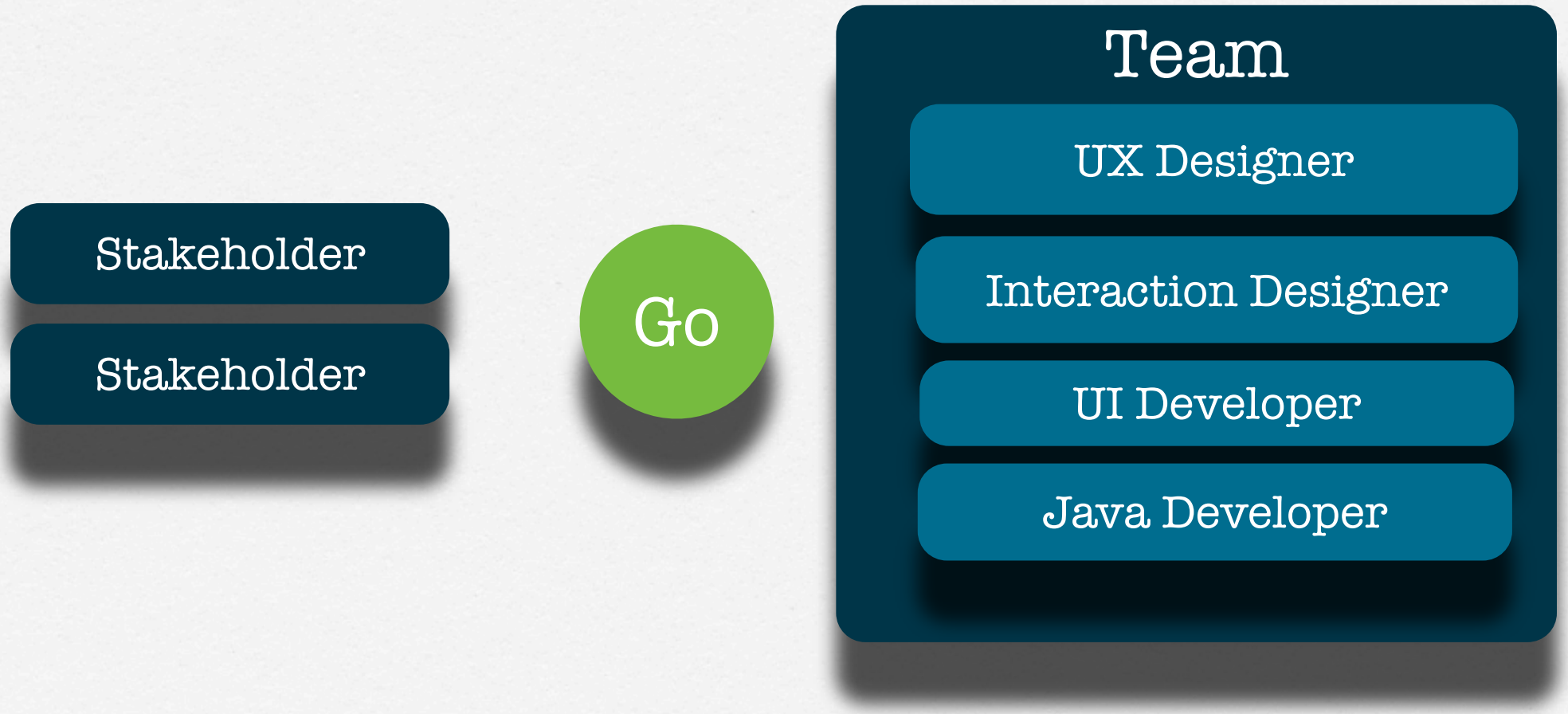
UI Dev

Java Developer Team

Rise of Client Side

- Web2.0, User focused, rapid development, GWT, GMail, Rich Client Apps, OpenSocial, 1000s of UI developers.
- Rapid Cycles

More balanced team



Scalability

- Light Server Load, mainly static content, lower bandwidth
- 1-4 CPU cores per user as opposed to 100's per server CPU
- Green & Cool ?

Efficient Development Model

- Server side is so slow to work with, weeks per app.
- Client side, fast cycles, days per app.

How development should happen

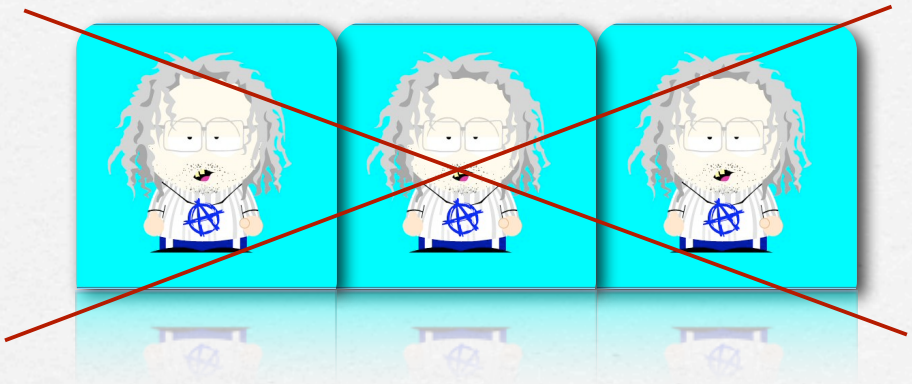
Java Developer Route

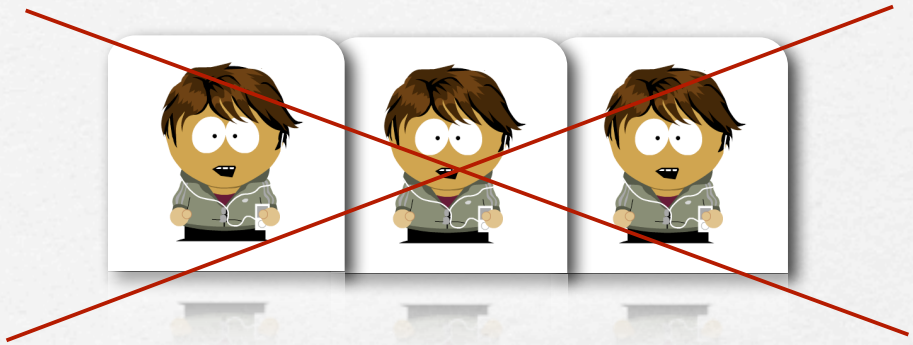
- Best-case: gather requirements
- Interpret requirements
- Long development process / Worst case: restart
- Quick testing interface
- Delivers final product
- Final: Lucky: perfect <--> Normally: out of range

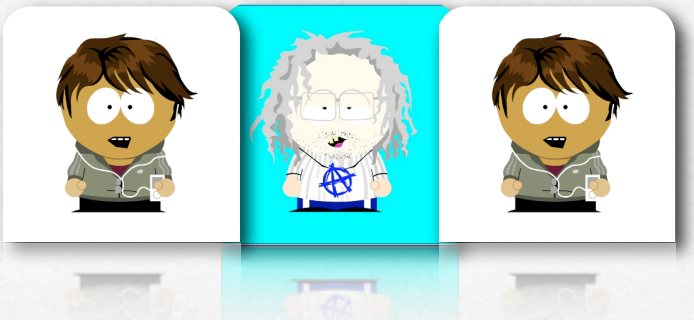
UI Developer Route

- Gather requirements, build use-cases, ...
- Interaction designs + wireframes
- Take to stakeholder and real users
- Adjust according to user feedback
- Repeat 3 and 4
- Translate: Data requirements => Java Developer
- Java Developer: can work data-oriented only and much more result focussed
- UI Developer develops wireframes and brings in functionality
- User feedback and adjustments during 8
- Final product: looks good, is functional, feels natural and represents what the user wants

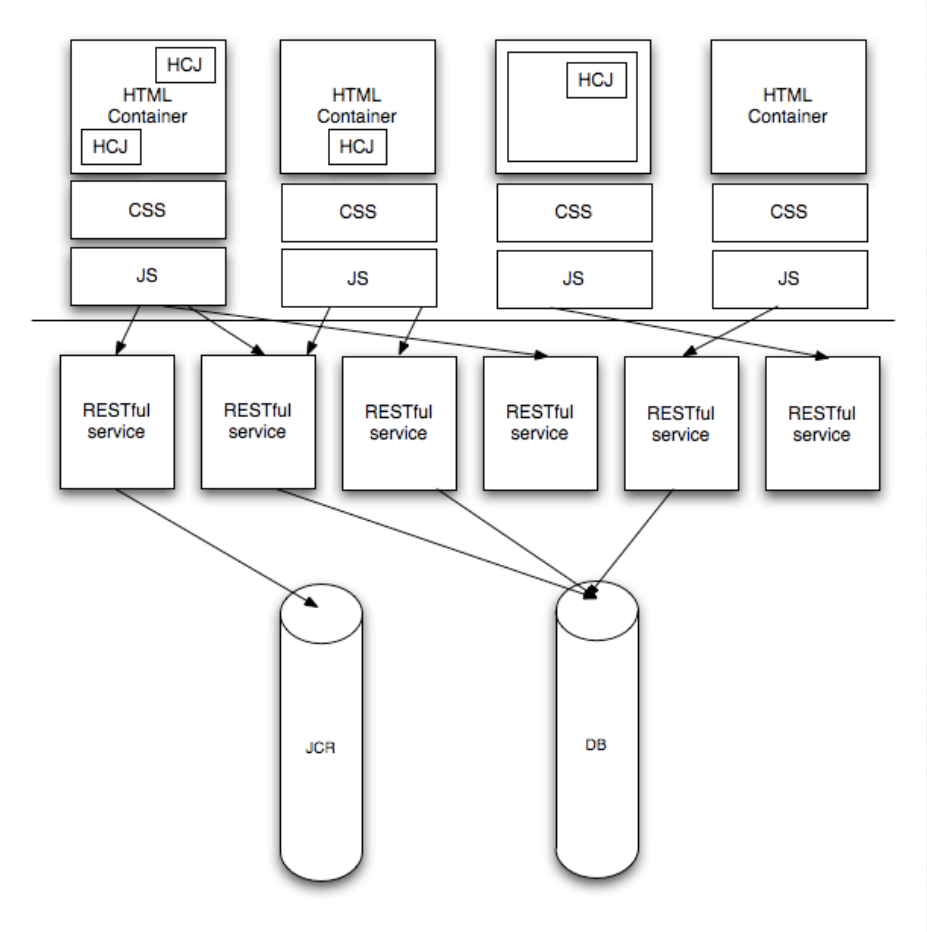
Team







Architecture



Back-end

- RESTful services
- Feed JSON data
- Work over the wire
- Follow the HTTP spec

JSON

JSON (pronounced"Jason"), short for JavaScript Object Notation, is a lightweight computer data interchange format. It is a text-based, human-readable format for representing simple data structures and associative arrays (called objects).

The JSON format is specified in RFC 4627 by Douglas Crockford. The official Internet media type for JSON is application/json. The JSON file extension is .json.

The JSON format is often used for transmitting structured data over a network connection in a process called serialization. Its main application is in Ajax web application programming, where it serves as an alternative to the use of the XML format.

Although JSON was based on a subset of the JavaScript programming language (specifically, Standard ECMA-262 3rd Edition—December 1999) and is commonly used with that language, it is considered to be a language-independent data format. Code for parsing and generating JSON data is readily available for a large variety of programming languages.

In December 2005, Yahoo! began offering some of its web services optionally in JSON. Google started offering JSON feeds for its GData web protocol in December 2006.

- http://en.wikipedia.org/wiki/JSON

JSON

- Associative array
- Easy to read, easy to process
- Small
- Evaluates straight into JavaScript object
- Nice library support

JSON

{"items":[{"motdBody":"Welcome to the Sakai conference in Paris","motdUrl":"http://localhost:8080/access/announcement/msg/!site/motd/4d023855-f537-42dd-afa1-7e11521d1279","motdTitle":"MOTD"}]}

JSON


```
{
    "items": [
        {
            "body": "Welcome to the Sakai conference in Paris",
            "url": "http://localhost:8080/access/announcement/msg/!site/motd/
4d023855-f537-42dd-afa1-7e11521d1279",
            "title": "MOTD"
        }
    ]
}
```

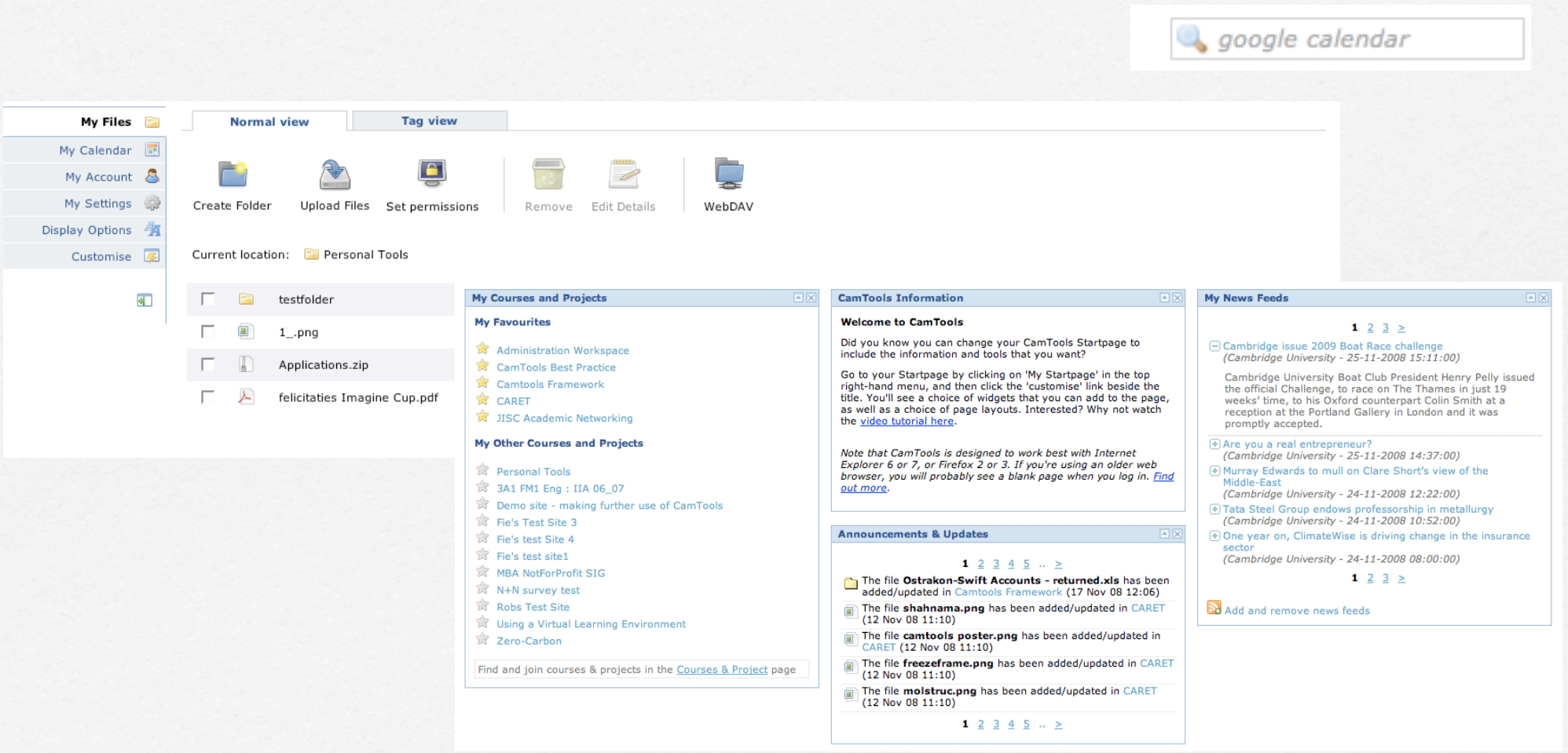
Front end

- Skeleton = set of plain HTML pages + CSS

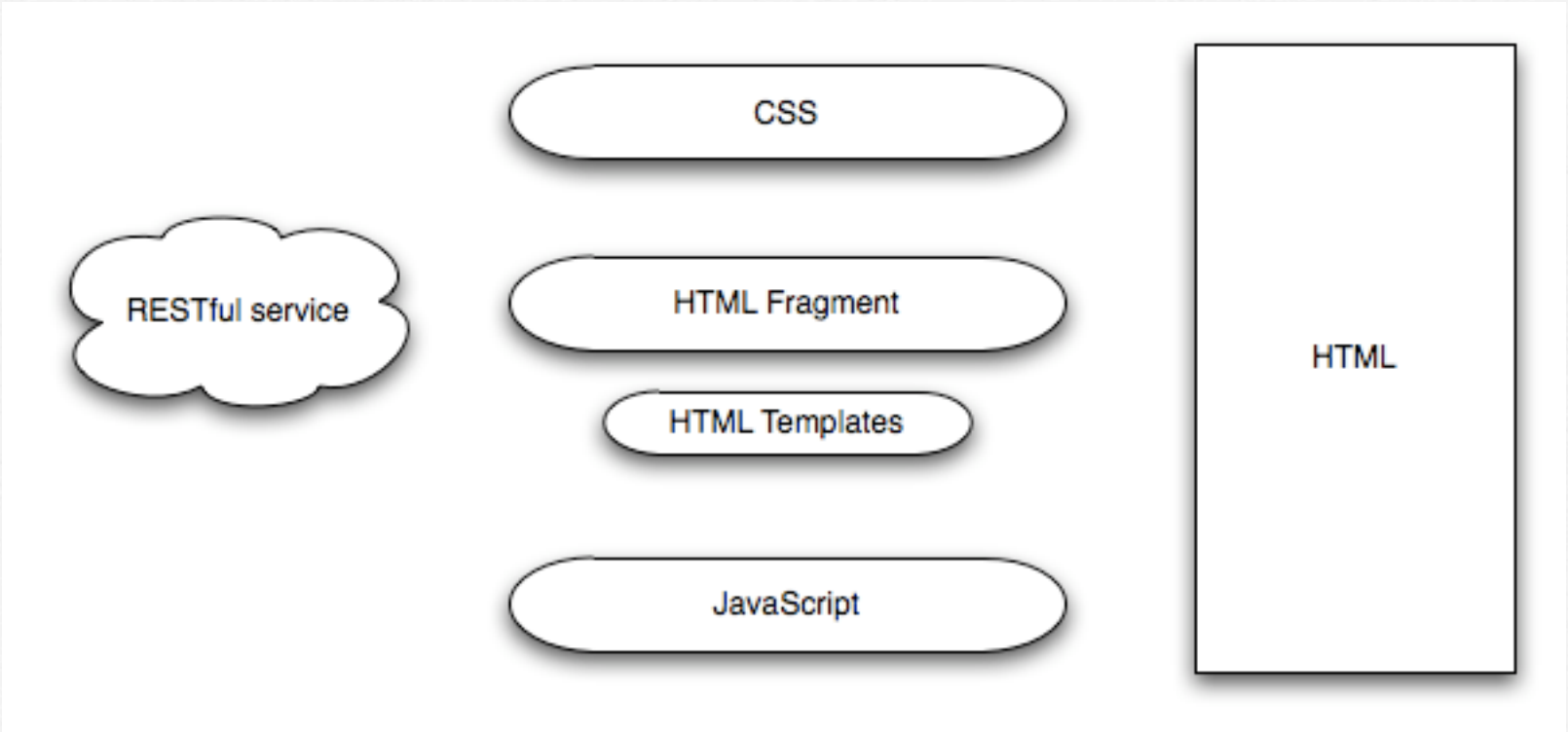
Widgets

- Place inside skeleton
- Reusability
- Multiple instances
- Modular

Widget Examples



Widget Architecture



Widgets specials

- Static files on disk
- SVN checkout into tomcat / webapps
- Load, change, save, refresh

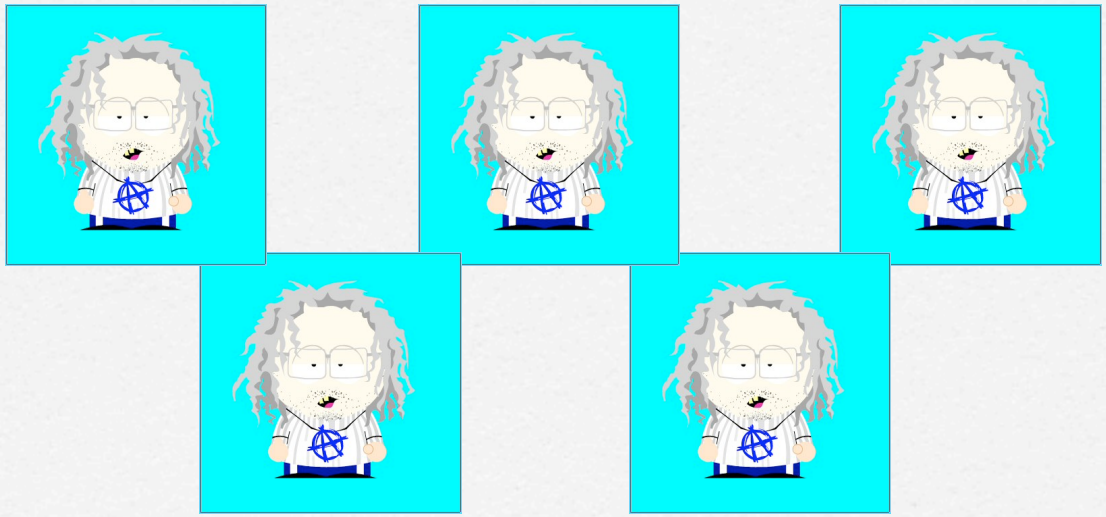
Widgets Specials

- Replacement for a tool
- No more iFrames !

JavaScript

- Key player

No !!!!!!!



JavaScript

- Why have we avoided JavaScript
- Why libraries ?
- Cross browser issues
- Browser Support

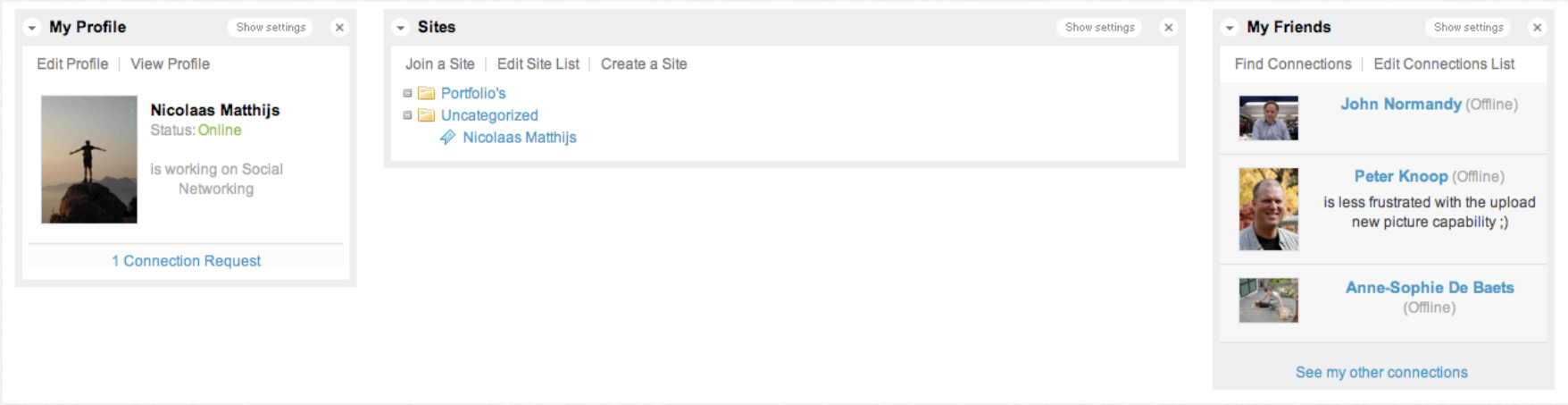
JavaScript

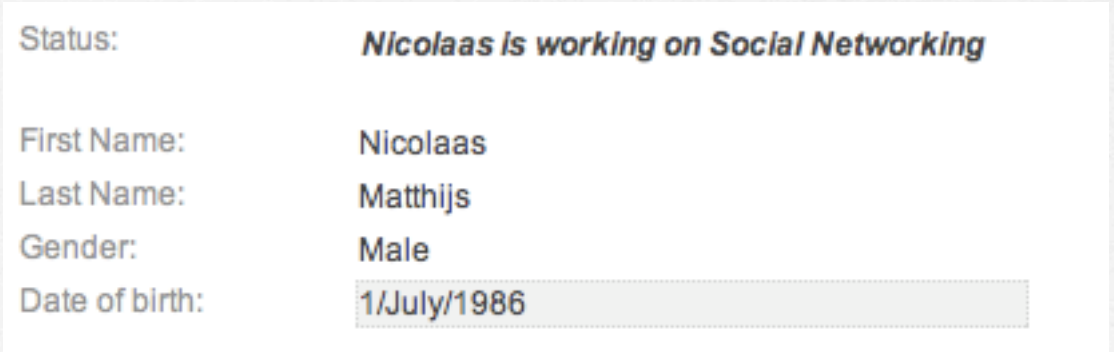
- => FEARLESS JAVASCRIPT
- Not hard any more
- Toolkits
- Documentation
- Future
- Speed

JavaScript

- Build and fills the screen
- Requests and receives
- Consumes JSON
- Allows very advanced interaction

JavaScript


My Profile
Show settings
Edit Profile | View Profile
Nicolaas Matthijs
Status: Online
is working on Social Networking
1 Connection Request
Sites
Show settings
Join a Site | Edit Site List | Create a Site
Portfolio's
Uncategorized
Nicolaas Matthijs
My Friends
Show settings
Find Connections | Edit Connections List
John Normandy (Offline)
Peter Knoop (Offline)
is less frustrated with the upload new picture capability ;)
Anne-Sophie De Baets
(Offline)
See my other connections


Status:
Nicolaas is working on Social Networking
First Name:
Nicolaas
Last Name:
Matthijs
Gender:
Male
Date of birth:
1/July/1986

JavaScript in Future

- iPhone, iTouch, ... support
- Development

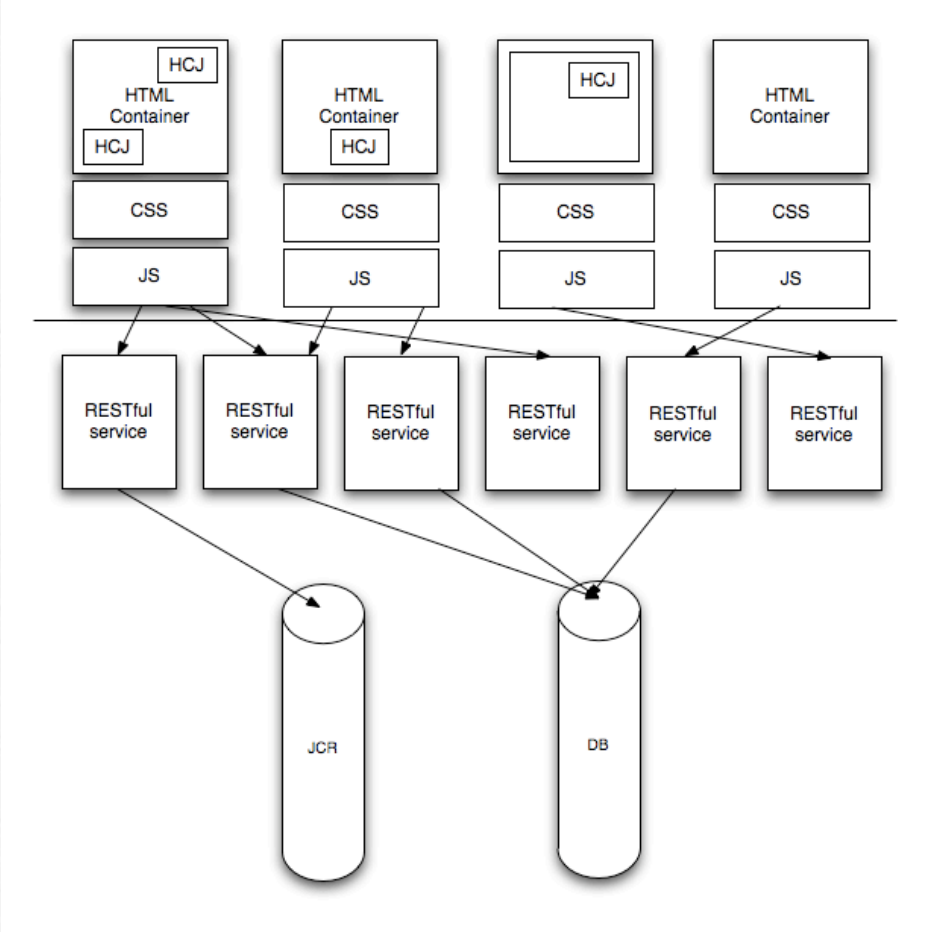
  http://webkit.org/blog/189/announcing-squirrelfish/

  http://code.google.com/p/v8/
- Accessibility support

JavaScript for me

- Learning is possible
- Trying is the best way
- Yes, we can help you

Architecture - Team



Theory is good

Evidence is better

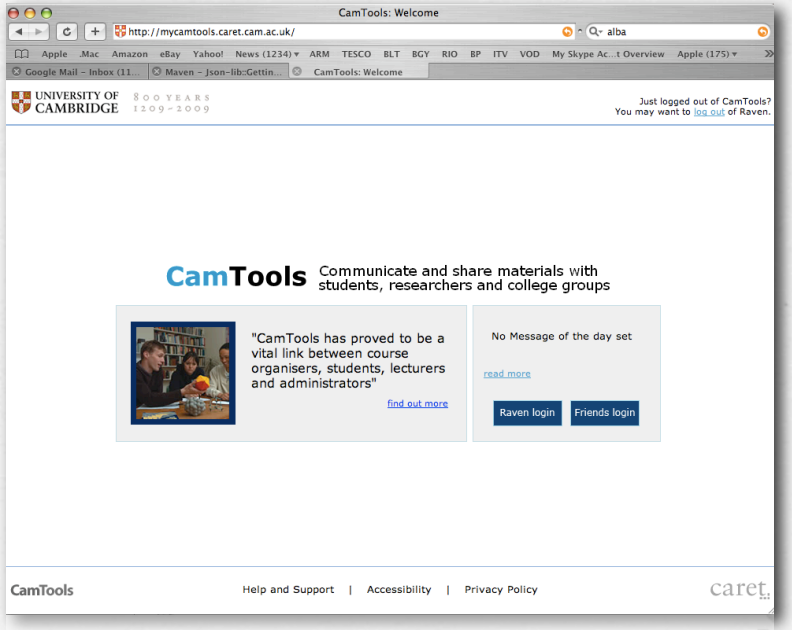
MyCamTools

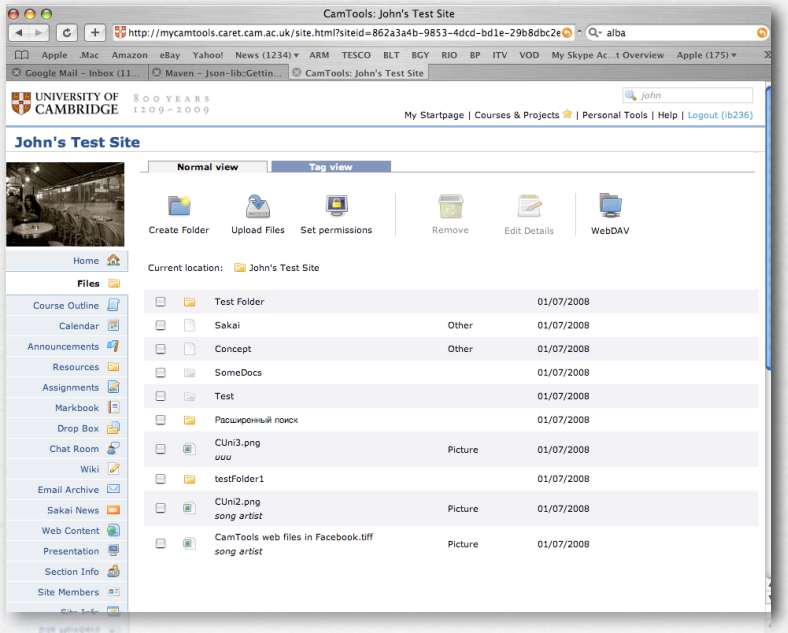
- Developed using this theory
- Full HTML - CSS - JavaScript front end
- Only static files
- Much progress
- In production since July

MyCamTools Goals

- Make our local Sakai more user friendly
- Lower development bar
- Speed up development
- Better seperation

Demo





http://camtools.cam.ac.uk

http://camtools-labs.caret.cam.ac.uk

Installing MyCamTools

- Sakai 2.5 or Sakai 2.6 installation
- Install and deploy Search
- search.enable=true (sakai.properties)
- Install and deploy JCR
- jcr.experimental=true (sakai.properties)

Installing MyCamTools

- Download and deploy SData
  https://source.caret.cam.ac.uk/camtools/trunk/camtools/sdata
- Go to TomCat webapps dir
- svn co https://source.caret.cam.ac.uk/camtools/trunk/camtools/flat/src/main/webapp ROOT
- svn co https://source.caret.cam.ac.uk/camtools/trunk/camtools/widgets/src/main/webapp widgets

UX Improvement Initiative

- Nathan Pearson (Sakai Foundation)
- => Make Sakai look better
- More info:
  http://bugs.sakaiproject.org/confluence/display/UX/Improving+the+CLE+1

UX Improvement Initiative

- Since August => Implemented most screens
- but ...
- kept making same assumptions

...

Sakai 3

Sakai 3 ?

- Ultimate goal: more useful + more user-centered
- Recycling of UX Initiative screens
- Break out tool silo's
- More flexible way of creating sites => context (cfr. Google Sites)

Sakai 3?

- Break down site as only organisational structure
- Central file repository => browsing, reusability, assessment, portfolio
- Creating ad-hoc non-site bound groups
- Social Networking

Timeline

- Now: design - prototype - implementation
- July 2008: Alpha release
- Production: Cambridge and GATech
- Help wanted !

Demo

http://sakai.sakaiproject.org

Installing Sakai 3

- Sakai 2.5 or Sakai 2.6 installation (for now)
- Install and deploy Search
- search.enable=true (sakai.properties)
- Install and deploy JCR
- jcr.experimental=true (sakai.properties)

Installing Sakai 3

- Download and deploy SData
  https://source.sakaiproject.org/contrib/tfd/trunk/sdata
- Go to TomCat webapps dir
- svn co https://source.sakaiproject.org/contrib/ux/branches/3.0/uxportal/src/main/webapp dev
- svn co https://source.sakaiproject.org/contrib/ux/branches/3.0/uxwidgets/src/main/webapp devwidgets

New !

- Develop Sakai without running it

```
$ apt-get install libapache2-mod-proxy-html

$ a2enmod proxy
$ # probably does this automatically: a2enmod proxy_http
$ # probably done by apt-get automatically: a2enmod proxy_html
$ a2enmod ssl


$ vim sites-enabled/000-default

# Must do this otherwise ProxyPass doesn't know what to do with https
SSLProxyEngine on

# Punch a hole through the proxying
ProxyPass /dev !
ProxyPass /devwidgets !
```

```
# Everything else goes through some running Sakai instance
ProxyPass / http://somerunninginstance
ProxyPassReverse / http://somerunninginstance

# Map the "hole" through to the filesystem
Alias /widgets.js "/home/raymond/ctlwidgets/htdocs/ROOT/widgets.js"
Alias /widgets/ "/home/raymond/ctlwidgets/htdocs/widgets/"


$ vim mods-enabled/proxy.conf

Add "Allow from all" (Debian defaults to paranoia)

$ /etc/init.d/apache2 restart
```

Hands-on: Frontend

JavaScript 101

Firebug (Firefox)

- Debugger
- Profiling tool
- DOM Inspector
- Interactive console

=> Your new best friend

http://www.get-firebug.com

Development Environment

- Aptana Studio
- JavaScript oriented version of Eclipse
- Intellicense
- jQuery support

Definitions

- DHTML = Dynamic HTML
- AJAX = Asynchronous JavaScript and XML
- RIA = Rich Internet Applications

But why bother at all ?

- [ ] Improved user experience
- [ ] Improved performance
- [ ] Expectations

Flash / SilverLight ?

- Political: Open vs Closed
- Practical: Open vs Closed
- Accessibility

DHTML/AJAX

- Open standards
- Works within the browser
- Accessibility

Part I: The Basics

- variables
- null vs undefined
- type coercion
- objects and arrays

Defining variables

- Define types with var
- No need to declare types

```
var mango = "yum";
mango = 12345;
```

Global variables

- If you ommit var --> global variable

```
rottenTomate = "gross" // This is global
```

Null vs Undefined

- null is the "nothing" value
- undefined is extremely nothing

Default value for defined variables
Error when JavaScript sees unknown variable

Null vs Undefined

```
var foo;
foo === undefined

console.debug(anothervariable); //undefined
```

Truthy and Falsey

- JavaScript does a lot of automatic type coercion
- Shades of true and false
- use careful - be aware
- useful when evaluating arguments

Falsey values

- false
- null
- undefined
- ""
- 0 (zero)
- NaN

- Everything else is truthy
  -1 , "false" , "0" are all true

Equal vs Equivalent

- Comparisons are coercive

  `1 == "1" // true`

  `0 == false // true`

- Non-coercive comparisons

  `1 === "1" // false`

  `0 === false // false`

Using Truthy and Falsey

- Checking for valid arguments before operating on them
- Substituting defaults
- Be careful with arguments that genuinely might be false, like numbers

```
if (apples){
    apples.eat();
}
```

Objects

Object are loose containers

- At their core : maps
- new Object() or {} returns empty container of key/value pairs
- Keys can be any string, values can be everything
- Two different ways to access members

```
basketOfFruit.apples // dot notation
basketOfFruit["bananas"] // subscript notation
```

- You can add new members to any object at any time

Object are modifiable

```
var basketOfFruit = {};

// New property
basketOfFruit.apples = "macintosh";

// New method
basketOfFruit.eat = function () {
 return “tasty”;
}
```

No Classes

- JavaScript doesn't have any concept of classes
- Methods are just properties in a container:
  - pass them around
  - modify them
  - delete them

First class functions

- Functions are data
- You can assign them
- You can pass them as arguments
- You can return them as results

Defining and using functions

```
var puree = function (aFruit) {
 return puree(aFruit);
};

function squeeze(aFruit) {
 return squeeze(aFruit);
}

function popsicle(juiceMakerFn, fruit) {
 var juice = juiceMakerFn(fruit);
 freeze(juice);
}
```

What does this mean

- You can pass bits of logic around and
- have them be invoked later
- Callbacks are easy to write and ubiquitous
- Functions are our basic building block

Don’t extend built in types

- Dynamic objects are awesome. But dangerous.
- Looseness allows us to change contracts for everyone
- Different scripts share the same browser window
- They all share the basic types
- Modifying built-in functionality will break things

Breaking built-in types

```
Object.prototype.keys = function () {
 var keys = [];
 for (prop in this) {
  keys.push(prop);
 }
 return keys;
}

var myKeys = {foo: "foo", bar: "bar"}.keys();
console.debug(myKeys); // [foo, bar, keys];
```

Constructor functions

- No classes in JavaScript, so how do we define new objects?
- Instantiate a function using the `new` keyword
- Any function can be used as a constructor
- Conventional to use `CamelCaseLikeThis`.

Constructor functions

```
function Apple(type, colour) {
  this.type = type;
  this.colour = colour;
};

var macintosh = new Apple("macintosh", "red");
```

Context and this

- JavaScript `this` pointer seems wild and unpredictable
- It points to different objects depending on the context (eg. global)
- Subtle, confusing, and powerful

that

Plain old functions and objects

```
// Just use plain old functions and objects.
function orange () {

    // Stable pointer to the current instance.
    var that = {};

    // Anything private stays inside here.
    // For public methods, just add properties.
    that.squeeze = function () {...}

    return that;

}
```

Closures

- Functions can be defined inside other functions
- Inner functions have access to the outer function's variables
- A closure is formed by returning the inner function from the outer function
- The inner function will still have access to all the variables from the outer function

Closure example

```
function addNumbers (a, b) {

 function addEmUp (c) {
  return a + b + c;
 }

 return addEmUp;
}

var add = addNumbers(1, 2); // result is an “add 3” Function
add(3); // Result is 6
add(5); // Result is 8
```

Toolkits

Foundational Toolkit

- Thousands of them ?
  - Single problem solutions
  - Widgets

JavaScript is hard ?

NO

This is hard

- browser inconsistencies and bugs
- complex data and user interfaces in web applications
- subtle and varied high-quality user interactions
- the call and response of asynchronous client-server interaction

Frameworks can help

- Browser Abstraction
- DOM traversal, selection, and manipulation
- easy and dynamic event binding
- quick, responsive, bullet-proof AJAX functionality

Criteria

- Cross-browser support
- Easy debugging
- Event abstraction
- A solid DOM manipulation library
- A strong community and clear roadmap for improvements
- Accessibility

jQuery

Write less - Do more

http://jquery.com

Finding something

- `jQuery("<selector>")`
- selectors =

  tags: `jQuery("tr")`
  ids: `jQuery("#myId")`
  classes: `jQuery(".myClass")`
  pseudo tags: `jQuery("div:first")`

Finding something

- `jQuery("<selector>")`
- more selectors = combining selectors

element by class: `jQuery("li.selected");`
relationships: `jQuery("tbody tr:even");`
children: `jQuery("div > p");`
siblings: `jQuery("div ~ p");`

etc, etc, etc...

Doing something

```
function stripeListElements(listId) {
 jQuery('#' + listId + " li:even").addClass("odd");
}
```

jQuery === $

Doing something

```
$(".some-hidden-thing").show();
$(".some-hidden-thing").fadeIn("slow");
$("<li>A new list item</li>").appendTo("#myList");
$("#myList li:last").replaceWith("<li>A new list item</li>");
$("div.container").clone().appendTo("body");
```

Doing something

```
$("div.container").clone().appendTo("body");
```

chaining...

```
$("div#mytemplate").addClass('menu')
    .clone()
        .appendTo("body")
            .click(dosomething());
```

Attaching events

```
$(".button").click(function(){
 doSomething();
});
$(".button").hover(function(){
 jQuery(this).addClass("hilite");
}, function(){
 jQuery(this).removeClass("hilite");
});
$(".button").focus(function(){
 jQuery(this).addClass("hilite");
});
$(".button").blur(function(){
 jQuery(this).addClass("hilite");
});
```

Online documentation

http://jquery.com

SData.js - Ajax

- Allows multipart file upload
- Has a login handler that checks logins

SData.js - Ajax

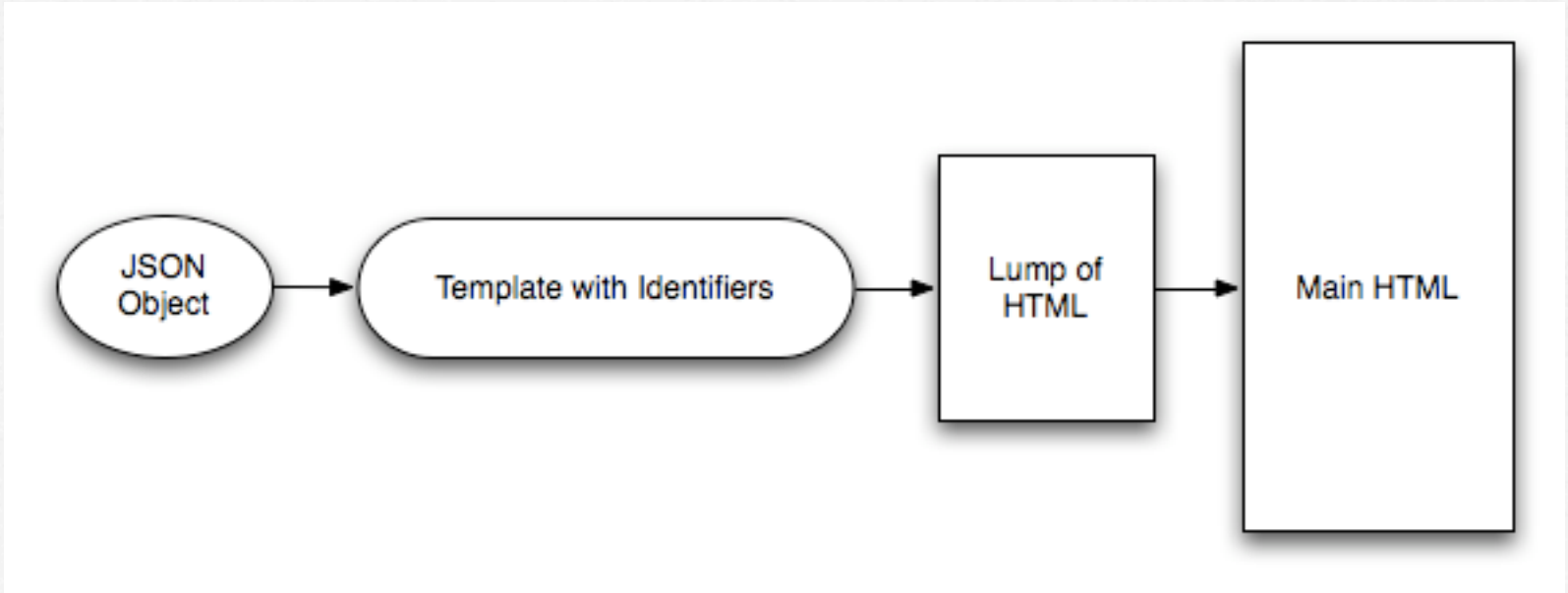
```
var parameters = {"sitename" : sitetitle,
"sitedescription" : sitedescription, "siteid" : siteid,
"type" : type };

sdata.Ajax.request({
      url :url,
      httpMethod : "POST",
      onSuccess : function(data) {
            renderList(newjson);
      },
      onFail : function(status) {
            alert("An error has occured");
      },
      postData : parameters,
      contentType : "application/x-www-form-urlencoded"
});
```

```
sdata.Ajax.request({
   httpMethod: "GET",
   url: "/sdata/me?sid=" + Math.random(),
   onSuccess: function(data){
       var me = eval('(' + data + ')');
       doSomething(me);
   },
   onFail: function(status){
      alert("An error has occured");
   }
});
```

SData.js - Templating

- Trimpath
- Similar to PHP smarty
- Easy changes



SData.js - Templating

```
<div id="sitelist_template" style="display:none"><!--
     <ul class="sites-menu">
          <li><a href="#">Uncategorized</a>
               <ul>
                    {for site in items}
                         <li><a href="/dev/site_home_page.html?siteid=${site.id}"
class="sites-menu-blue">${site.title}</a></li>
                    {/for}
               </ul>
          </li>
     </ul>
--></div>
```

```
var renderList = function(json){
     $("#sitelist",rootel).html(sdata.html.Template.render('sitelist_template', json));
}
```

QueryString.js

- Read QueryString parameters

```
var qs = new Querystring();
var siteId = qs.get("site", false);

if (siteId) {
   siteJsonUrl = "/direct/site/" + siteId + ".json";
} else {
   alert(“invalid site id”);
}
```

Others

- jQuery UI (http://ui.jquery.com)
- SWFUpload (http://swfupload.org)
- RSH (http://code.google.com/p/reallysimplehistory/)
- Others ...

JavaScript a11y

DHTML

- The shift from documents to applications
- Familiar a11y techniques aren't enough
- Most DHTML is completely inaccessible
- New techniques are still being figured out

The problem

- Custom widgets often look, but don’t act, like their counterparts on the desktop
- HTML provides only simple semantics
- Not enough information for ATs
- Dynamic updates require new design strategies to be accessible

The solution

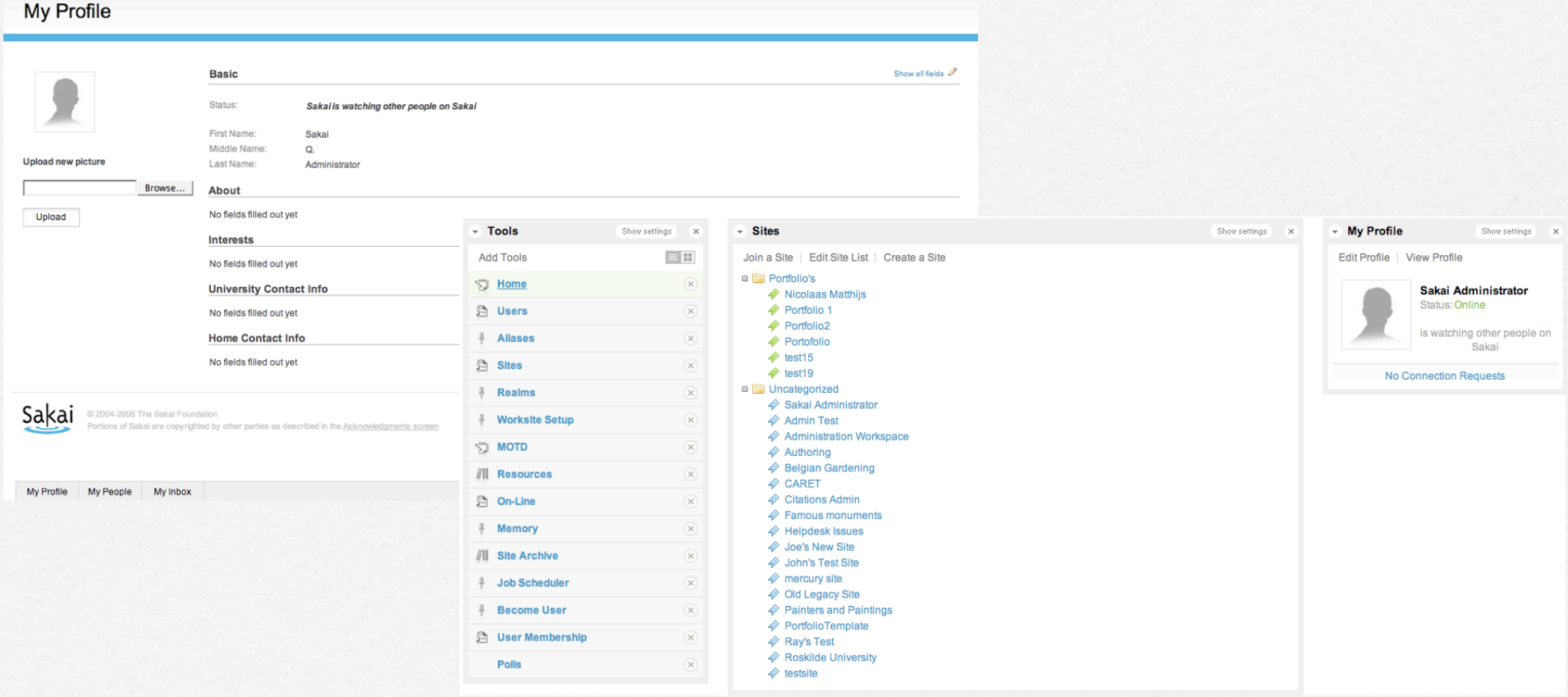
- Describe user interfaces with ARIA
- Add consistent keyboard controls
- Provide flexible styling and presentation

=> Fluid Infusion

Status

- ARIA standard (in progress)
- KeyBoard accessibility
  - => Fluid infusion
  - => jQuery keyboard a11y plugin
- Provide flexible styling and presentation
  - => Just started off

Keyboard a11y



Keyboard conventions

- Tab key focuses the control or widget
- Arrow keys select an item
- Enter or Spacebar activate an item

- Tab is handled by the browser. For the rest, you need to write code.

Tabbing and tabindex

- Each focusable item can be reached in sequence by pressing the Tab key
- Shift-Tab moves backwards
- The `tabindex` attribute allows you to customize the tab order
- `tabindex="-1"` removes element from the tab order: useful for custom handlers

Setting Tabindex

```
// Put the tab list in the tab order.
jQuery(“#animalTabs”).tabindex(0);

// Remove the individual tabs from the tab order.
// We’ll focus them programmatically with the arrows.
jQuery(“#animalTabs li”).tabindex(-1);
```

Navigating with arrow keys

```
// Make the tabList focusable with Tab.
var tabList = jQuery(“#animalTabs”).tabbable();

// Make the tabs selectable with the arrow keys.
var tabs = jQuery(“li”, tabList);
tabs.selectable(tabList, {
  willSelect: function(aTab) {
    aTab.addClass(“highlight”);
  }
});
```

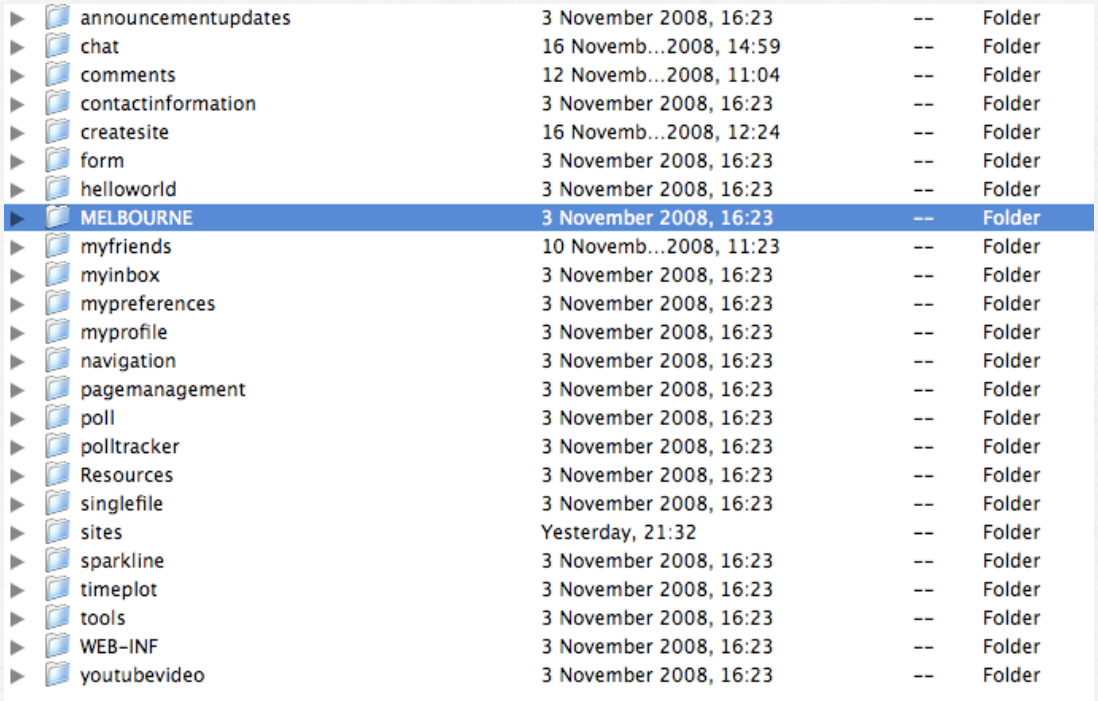
Activation handlers

```
// Make each tab activatable with Enter & Spacebar
tabs.activatable(function(aTab) {
  alert(“You just selected: “ + aTab.text());
});
```

CheckList

CheckList

- [ ] Create new folder



announcementupdates	3 November 2008, 16:23	--	Folder
chat	16 Novemb...2008, 14:59	--	Folder
comments	12 Novemb...2008, 11:04	--	Folder
contactinformation	3 November 2008, 16:23	--	Folder
createsite	16 Novemb...2008, 12:24	--	Folder
form	3 November 2008, 16:23	--	Folder
helloworld	3 November 2008, 16:23	--	Folder
MELBOURNE	3 November 2008, 16:23	--	Folder
myfriends	10 Novemb...2008, 11:23	--	Folder
myinbox	3 November 2008, 16:23	--	Folder
mypreferences	3 November 2008, 16:23	--	Folder
myprofile	3 November 2008, 16:23	--	Folder
navigation	3 November 2008, 16:23	--	Folder
pagemanagement	3 November 2008, 16:23	--	Folder
poll	3 November 2008, 16:23	--	Folder
polltracker	3 November 2008, 16:23	--	Folder
Resources	3 November 2008, 16:23	--	Folder
singlefile	3 November 2008, 16:23	--	Folder
sites	Yesterday, 21:32	--	Folder
sparkline	3 November 2008, 16:23	--	Folder
timeplot	3 November 2008, 16:23	--	Folder
tools	3 November 2008, 16:23	--	Folder
WEB-INF	3 November 2008, 16:23	--	Folder
youtubevideo	3 November 2008, 16:23	--	Folder

Checklist

- [ ] Make the right directories

Checklist

- [ ] Fill in HTML

```
<div id="melbourne_settings" style="display:none">

</div>
<div id="melbourne_output" style="display:none; border: 1px solid #CCCCCC; width:400px; padding: 5px">

</div>

<script type="text/javascript" language="JavaScript" src="/devwidgets/MELBOURNE/javascript/
melbourne.js"></script>
```

Checklist

- [ ] Fill in JavaScript

```
var sakai = sakai || {};

sakai.melbourne = function(tuid, placement, showSettings){

      // To Do: Code

};

sdata.widgets.WidgetLoader.informOnLoad("melbourne");
```

Checklist

- [ ] Register + Define

```
melbourne :
{
      description:"MELBOURNE",
      iframe:0,
      url:"/devwidgets/MELBOURNE/melbourne.html",
      name:"Melbourne",
      id:"melbourne",
      personalportal:0,
      siteportal:0,
       ca:1,
       img:"/devwidgets/MELBOURNE/images/logo_inst.gif"
   }
```

Checklist

- [ ] Get started

```
var sakai = sakai || {};
sakai.melbourne = function(tuid, placement, showSettings){

	var rootel = $("#" + tuid);

	var loadSettings = function(){
		$("#melbourne_settings").show();
		$("#melbourne_settings").html("Showing settings");
	}
	var showPreview = function(){
		$("#melbourne_output").show();
		$("#melbourne_output").html("Showing settings");
	}

	if (showSettings){
		loadSettings();
	} else {
		showPreview();
	}
};

sdata.widgets.WidgetLoader.informOnLoad("melbourne");
```

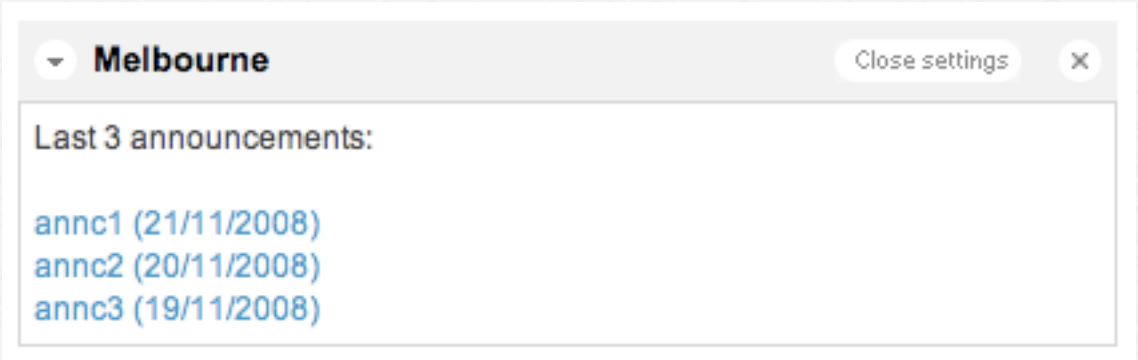
Checklist

- [ ] AJAX example

```
sdata.Ajax.request({
   url :"/sdata/f/" + placement + "/" + tuid + "/configuration?
   sid=" + Math.random(),
   httpMethod : "GET",
   onSuccess : function(data) {
      var json = eval('(' + data + ')');
      numberToDisplay = json.number;
      getAnnouncements();
   },
   onFail : function(status) {
      showVideo(status,false);
      getAnnouncements();
   }
});
```

Example

- Last x announcements widgets



Q & A