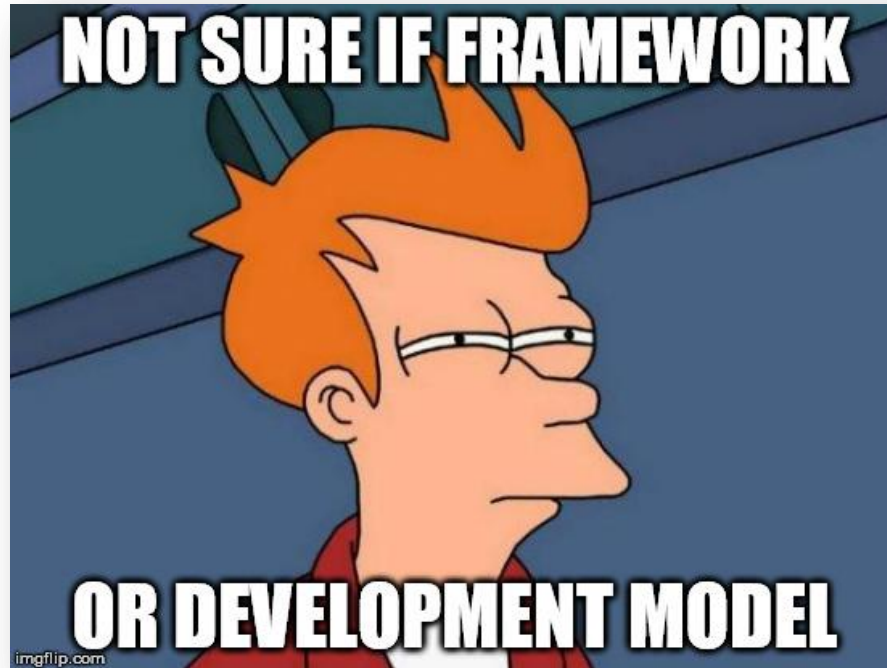# 2 years of Aikau

How is it faring and what do we want from it?

Axel Faust, Senior IT-Consultant / Software Architect

What is Aikau?

NOT SURE IF FRAMEWORK
OR DEVELOPMENT MODEL

What is Aikau?

- „Just" a library
  - Widgets / services
  - Tests suite during build

- Dev / customization model
  - Modular components
  - Publication / subscription
  - Declarative page model
  - Post-processing
  - Dependency collection / aggregation

Frameworks behind the „framework"

(to a limited extent)

BeeCon 2016

What is Aikau?
(Very simple) Examples

```
var services, widgets;

widgets = [{
    name : 'alfresco/lists/AlfList',
    config : {
        loadDataPublishTopic : 'LOAD_SIMPLE_LIST',
        widgets : [{
            name : 'alfresco/lists/views/AlfListView',
            config : {
                widgets : [{
                    name : 'alfresco/lists/views/layouts/Row',
                    config : {
                        widgets : [{
                            name : 'alfresco/lists/views/layouts/Cell',
                            config : {
                                widgets : [{
                                    name : 'alfresco/renderers/Property',
                                    config : {
                                        propertyToRender : 'id'
                                    }
                                }]
                            }
                        }]
                    }
                }]
            }
        }]
    }
}];

services = ['test/SimpleListService'];

model.jsonModel = {
    rootNodeId : 'simpleList',
    services : services,
    widgets : widgets
};
```

Modular services / widgets

```
define(["dojo/_base/declare", "dijit/_WidgetBase", "dijit/_TemplatedMixin", "dojo/text!./templates/Label.html",
        "alfresco/core/Core", "dojo/_base/lang", "dojo/dom-construct"],
    function(declare, _WidgetBase, _TemplatedMixin, template, AlfCore, lang, domConstruct) {
    return declare([_WidgetBase, _TemplatedMixin, AlfCore], {

        i18nRequirements: [{i18nFile: "./i18n/Label.properties"}],
        cssRequirements: [{cssFile:"./css/Label.css"}],
        templateString: template,                                               rvice, lang)

        label: "",
        subscriptionTopic: null,
        subscriptionPayloadProperty: "label",

        postMixInProperties: function alfresco_html_Label__postMixInProperties() {
            this.label = this.message(this.label);
            if (this.subscriptionTopic && lang.trim(this.subscriptionTopic) !== "")
            {
                this.alfSubscribe(this.subscriptionTopic, lang.hitch(this, this.onLabelUpdate));
            }
        },

        onLabelUpdate: function alfresco_html_Label__onLabelUpdate(payload) {
            var update = lang.getObject(this.subscriptionPayloadProperty, false, payload);
            if (update !== null && typeof update !== "undefined")
            {
                domConstruct.empty(this.labelNode);
                var textNode = document.createTextNode(update);
                this.labelNode.appendChild(textNode);
            }
        }
    });
});
```

BeeCon 2016

Aikau milestones

- Feb 2013: first public SVN code
- Aug 2013: header model 4.2.d

- Jan 2014: „Introducing Aikau"
- May 2014: faceted search

- Jan/Feb 2015: GitHub / separation from Share
- Mar 2015: Alfresco 5.0.d + updateable Aikau

- April 2016: > 60 weekly releases

BeeCon 2016

2 years of Aikau

Use cases from trivial to complex

Simple customizations
Example: Faceted search result

Simple customizations

- Pretty straightforward
  - Extension module + custom JS
  - widgetUtils for traversal / manipulation
  - IDs from base JS file

- See Aikau jsdoc for widgets / params
  http://dev.alfresco.com/resource/docs/aikau-jsdoc
  - Follow extends/mixes for inherited params

```
var headerMenuBar, widgets = model.jsonModel.widgets;

widgetUtils.deleteObjectFromArray(widgets, 'id', 'HEADER_MY_FILES');
widgetUtils.deleteObjectFromArray(widgets, 'id', 'HEADER_SHARED_FILES');
widgetUtils.deleteObjectFromArray(widgets, 'id', 'HEADER_SITES_MENU');
widgetUtils.deleteObjectFromArray(widgets, 'id', 'HEADER_TASKS');
widgetUtils.deleteObjectFromArray(widgets, 'id', 'HEADER_PEOPLE');
widgetUtils.deleteObjectFromArray(widgets, 'id', 'HEADER_REPOSITORY');

headerMenuBar = widgetUtils.findObject(widgets, 'id', 'HEADER_APP_MENU_BAR');

headerMenuBar.config.widgets.splice(1, 0, {
    id : 'HEADER_POC_ORG_CONFIG',
    name : 'alfresco/menus/AlfMenuBarItem',
    config : {
        label : 'page.poc-supplier-meta.title',
        targetUrl : '<url>'
    }
});
```

```
  var searchHeader = widgetUtils.
      findObject(model.jsonModel.widgets, 'id', 'HEADER_SEARCH');
  searchHeader.name = 'prodyna-search/header/SearchBox';
  searchHeader.config.popupMenuWidgets =
      searchHeader.config.popupMenuWidgets || [];
  if (page.url.templateArgs.site)
  {
      searchHeader.config.popupMenuWidgets.splice(0, 0,
      {
          name : 'prodyna-search/header/SearchInputAwareMenuItem',
          config :
          {
              label : '<label>',
              targetUrl : '<url>',
              inputAwareTargetUrl : '<url>'
          }
      });
  }
```

BeeCon 2016

Simple customizations
Limitations / issues

- Customizability of (new) pages
  - E.g. no widget IDs in faceted search until 5.0.2
  - Changes to key widgets + services

- Customizability of complex / composite widgets
  - Partial / no JSON sub-model

- JSDoc = terse + developer-focussed + unversioned

documentAndFolderActions :string[]

This can be configured to override the default filter for the actions that are applicable to folder and document nodes. Actions need to be filtered as Aikau does not currently support all of the actions that can be configured in Alfresco Share. However, if custom actions are provided by an extension or further filtering is required then they can be configured using this attribute.

Type:
- string[]

Default Value: null
Source: search/AlfSearchResult.js, line 124

Custom features / enhancements
Example: Form enrichment
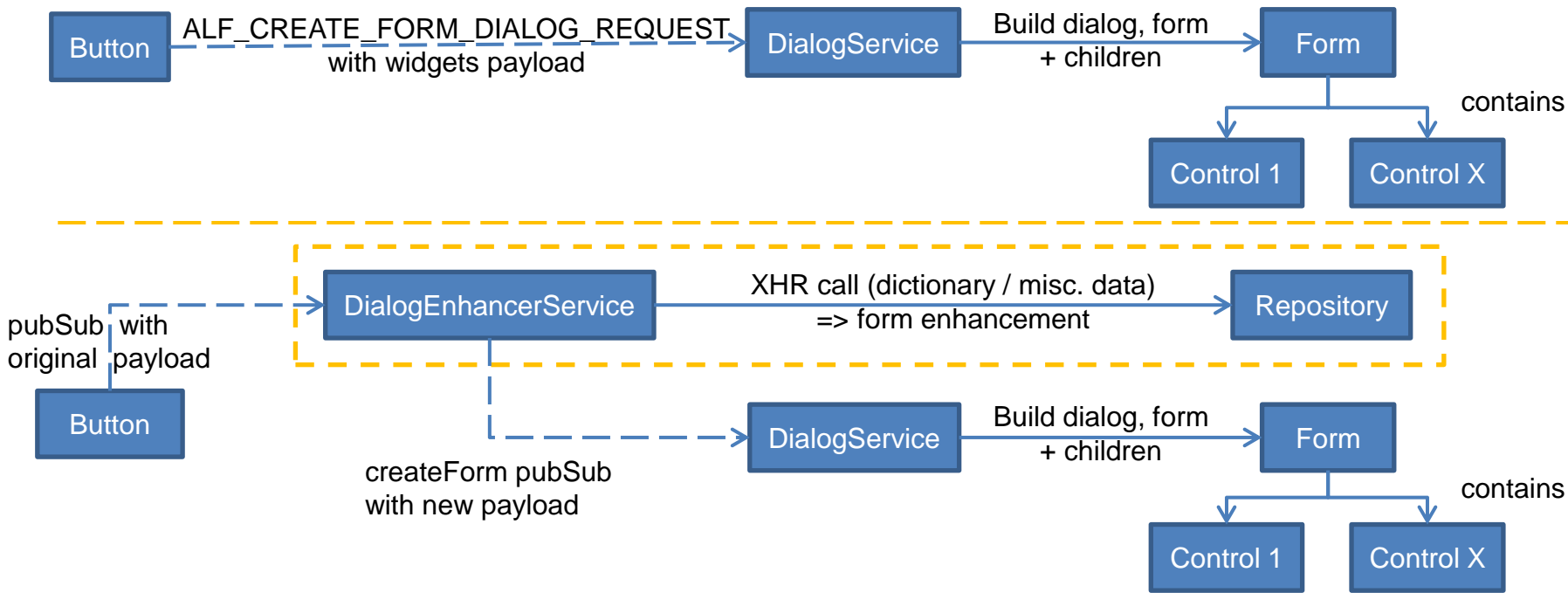
Custom features / enhancements

- Dozens of „starter" modules / mixins
  - WidgetBase + Templated
  - AbstractService (1.0.32+)
  - Core + CoreXhr

- Cornerstone: pubSub
  - Widget ⇔ service
  - „Request" + „Response"

- Complex user interactions
  - Small services w/ distinct responsibility
  - pubSub chaining / orchestration

```javascript
define([ 'dojo/_base/declare', 'alfresco/services/BaseService', 'dojo/_base/lang' ],
    function(declare, BaseService, lang)
{
    return declare([ BaseService ], {

        generatedRowCount : 25,

        registerSubscriptions : function test_SimpleListService__registerSubscriptions()
        {
            this.alfSubscribe('LOAD_SIMPLE_LIST', lang.hitch(this, this.onLoadSimpleList));
        },

        onLoadSimpleList : function test_SimpleListService__onLoadSimpleList(payload)
        {
            var idx, dataEntries = [];

            for (idx = 0; idx < this.generatedRowCount; idx++)
            {
                dataEntries.push({
                    id : idx
                });
            }

            this.alfPublish((payload.alfResponseTopic || 'LOAD_SIMPLE_LIST') + '_SUCCESS', {
                items : dataEntries,
                metadata : {}
            }, false, false, payload.alfResponseScope || '');
        }
    });
});
```

BeeCon 2016

Custom features / enhancements
Limitations / issues

- Learning curve for „citizen developers"
  - AMD, pubSub, „how module X works"
  - Tutorials partially outdated

    „I hate subscription model already"   *„looks like I start loving YUI2"*

- Limited ability to redirect / intercept standard pubSub
  - (Virtually) hard-coded topics / constants
  - Service sub-classing problematic

- Few defined extension points (1.0.60 => 18)
  - Only config params guaranteed

Custom pages / apps
Example: Complex page prototype

Navigation into different sub-apps
(i.e. master data management, dynamically loaded menu contents)

Dynamic tabs with simple or
semi-complex controls
(secondary lists; DMS)

- Nested, semi-scrollable data list
- complex inline editing
- Up to 200 dynamic columns
- bulk export / import

BeeCon 2016

Custom pages / apps

- Common use case modules / mixins
  - DnD, upload, charts, a11y, prototyping…

- Easy to add productivity tooling
  - Runtime model processing / generation
  - Modelling + model-to-text transformation

Custom pages / apps
Limitations / issues

- Ramp-up and productivity
  - Learning curve + dev optimization

- Performance and scalability
  - Heap (ACE-5049), page load (ACE-4812), rendition times (Aikau #958)

- Missing key features
  - Forms framework / doclib

- Rich feature extensions vs. easy lib update

2 years of Aikau

Where are we at / What do we want?

Where are we at?

- GitHub
  - 3.300+ commits, 80+ releases, 15 contributors (primarily Alfresco)
  - Pillars of Aikau: Dave Draper, Martin Doyle

- Product
  - 5 active + 4 prototype pages + header component
  - Individual components in RM and other (?)

- Community
  - 1 „real" project (https://github.com/EisenVault) + several tutorial runs
  - (Partially) adopted by 3 – 5 in #alfresco
  - Unknown adoption by customers / partners

Discussing low adoption within the community
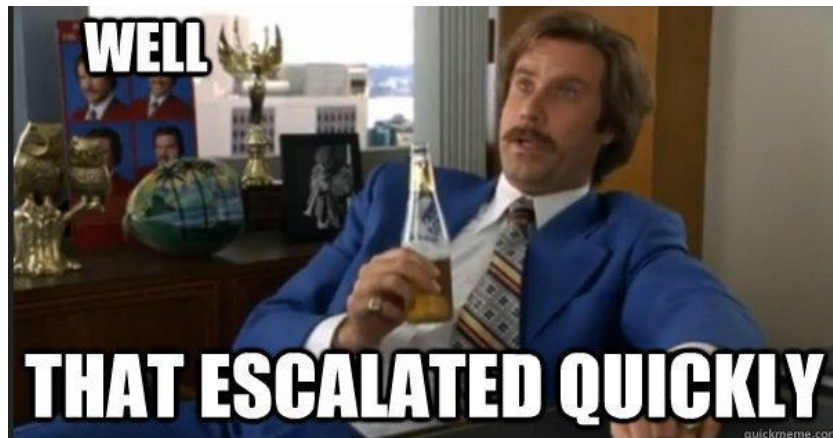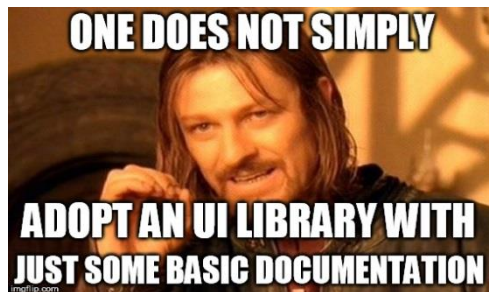
What do we want?

- Clear strategy / timeframe for „updated Share UI"
  - Defined lib + target of X years
  - Eliminate YUI / competing UI libs

- Priority for key functionality / components
  - Who adopts niche UI?
  - Find + fix scalability issues early/earlier

- Extended documentation
  - Architectural / design concepts
  - Implied / expected conventions

Try & adopt Aikau
Provide feedback / requests
(Build utilities if necessary)

Consistent + flexible UI layer
in 5.3 / 6.0?

Dismiss Aikau / discourage its use
Use / push alternative UI library

Increased UI fragmentation
Repeat of Aikau growing pains?
Share replaced with multiple,
isolated clients?

BeeCon 2016

I, for one, welcome our (new) Aikau overlord

* Edward Ryon Makuahanai Aikau

No alternatives with net benefits

Conceptionally sound

Org. / tech. Issues easy to address

Thank you

BeeCon 2016