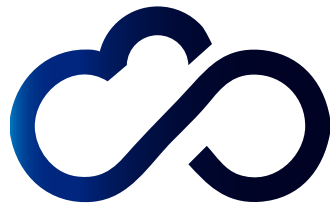


meet the new guys: **dRest & dFace**



INDIGO DataCloud

Olufemi S. Adeyemi
Marina Sahakyan

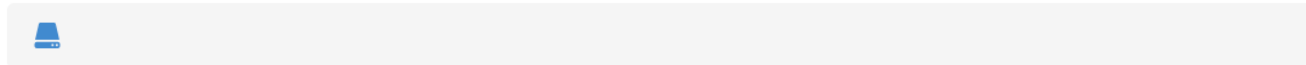
On behalf of the project team



The Old Guy

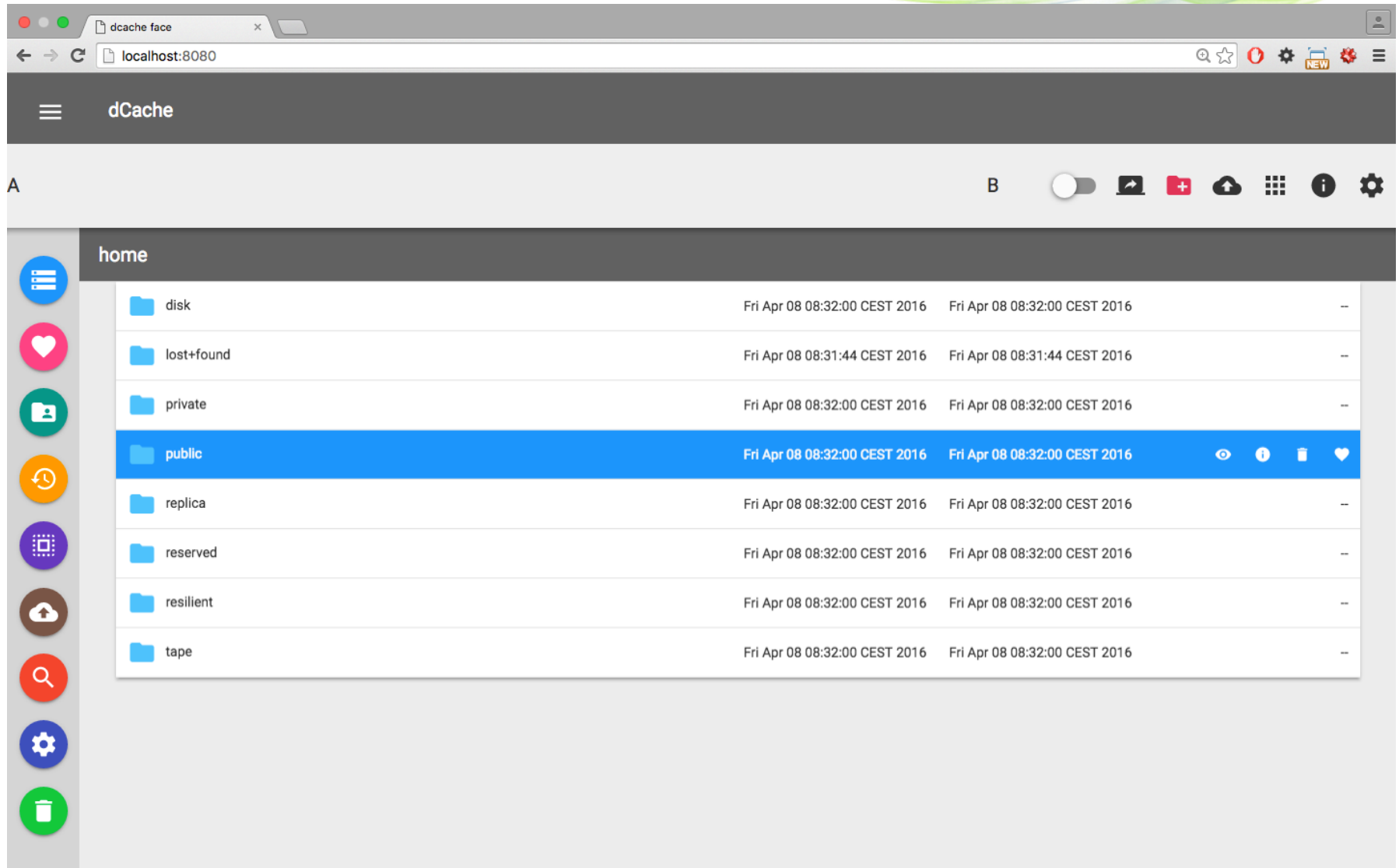
← → ↻  localhost:2880       

dCache system-test (built from 552fc76) 



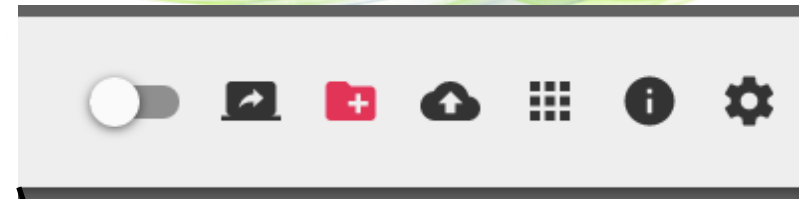
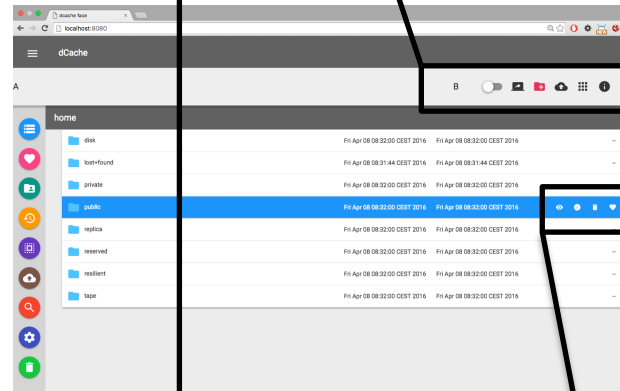
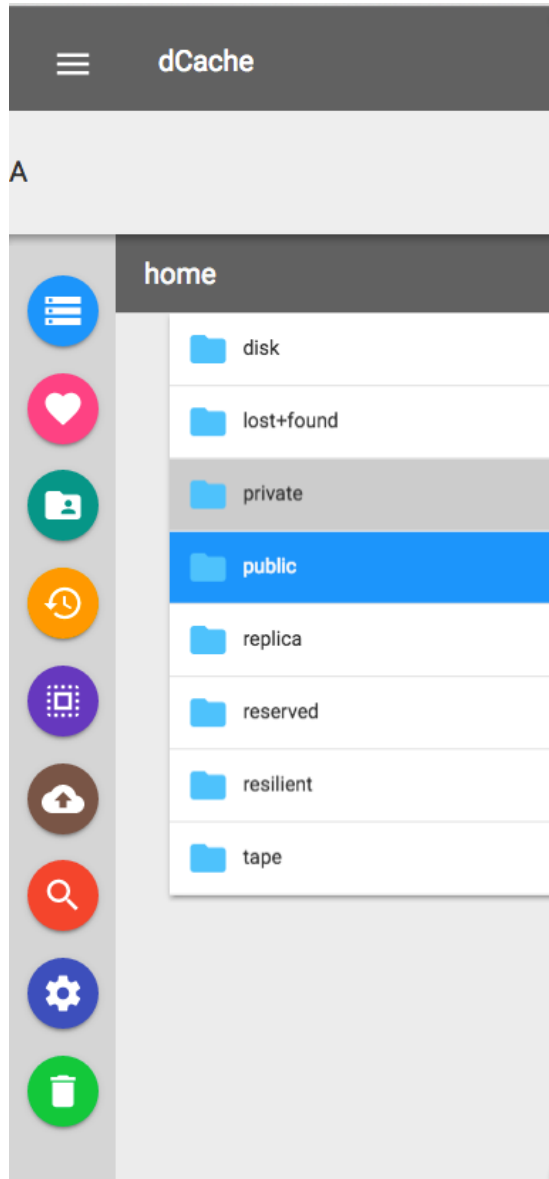
	Name	↕	Size	↕	Last Modified	↕
	disk				Sat Apr 09 13:04:02 CEST 2016	
	lost+found				Sat Apr 09 13:03:46 CEST 2016	
	private				Sat Apr 09 13:04:02 CEST 2016	
	public				Sat Apr 09 13:04:02 CEST 2016	
	replica				Sat Apr 09 13:04:02 CEST 2016	
	reserved				Sat Apr 09 13:04:02 CEST 2016	
	resilient				Sat Apr 09 13:04:02 CEST 2016	
	tape				Sat Apr 09 13:04:02 CEST 2016	

Powered by [dCache](#)



The screenshot shows the dFace web interface in a browser window. The address bar indicates the URL is localhost:8080. The interface has a dark grey header with the 'dCache' logo and a sidebar on the left with various navigation icons. The main content area displays a list of folders under the 'home' directory. The 'public' folder is highlighted in blue.

Folder Name	Created	Modified	Actions
disk	Fri Apr 08 08:32:00 CEST 2016	Fri Apr 08 08:32:00 CEST 2016	--
lost+found	Fri Apr 08 08:31:44 CEST 2016	Fri Apr 08 08:31:44 CEST 2016	--
private	Fri Apr 08 08:32:00 CEST 2016	Fri Apr 08 08:32:00 CEST 2016	--
public	Fri Apr 08 08:32:00 CEST 2016	Fri Apr 08 08:32:00 CEST 2016	   
replica	Fri Apr 08 08:32:00 CEST 2016	Fri Apr 08 08:32:00 CEST 2016	--
reserved	Fri Apr 08 08:32:00 CEST 2016	Fri Apr 08 08:32:00 CEST 2016	--
resilient	Fri Apr 08 08:32:00 CEST 2016	Fri Apr 08 08:32:00 CEST 2016	--
tape	Fri Apr 08 08:32:00 CEST 2016	Fri Apr 08 08:32:00 CEST 2016	--



Objective

- ☑ *create a user-friendly web application with all the necessary User Interactions*
- ☑ *replace the current **dCache webdav** html pages*
- ☑ *integrate the user interface with dCache through “webdav door”*
- ☑ ***easily** maintained from both user and developer point of view*

Memory Lane

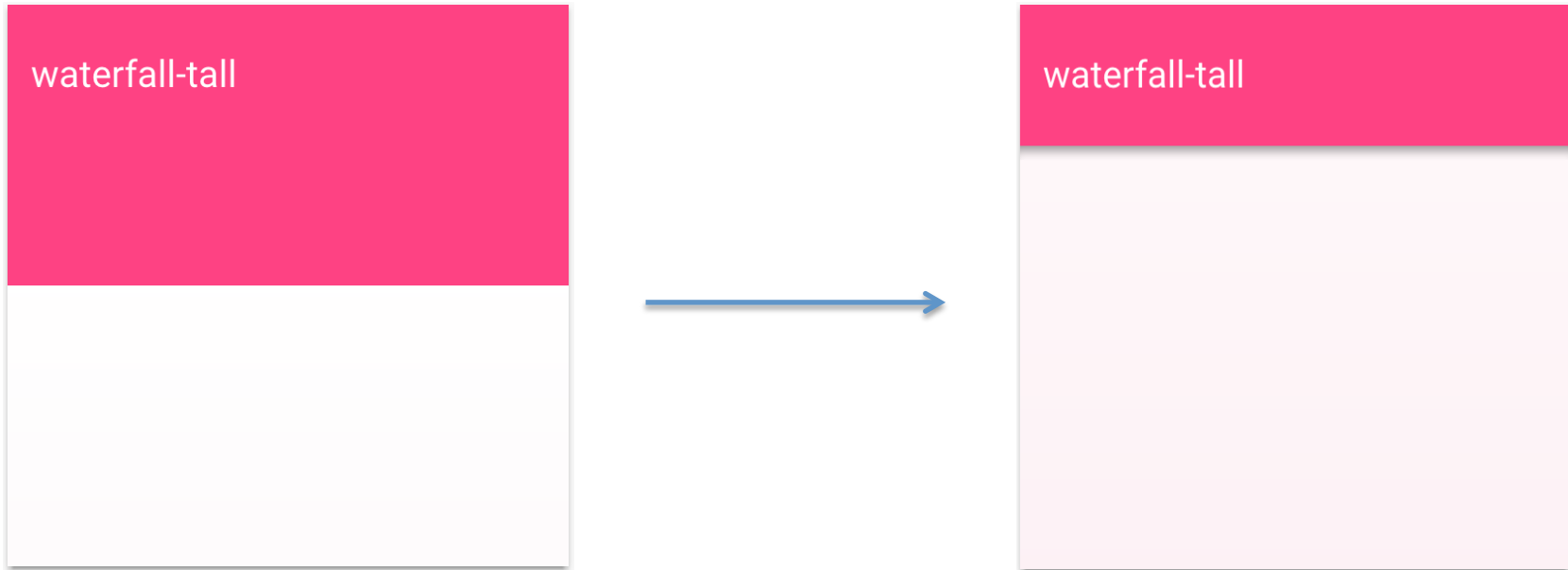
- HTML provides a set of built-in element like:

`<button>`, `<form>`, `<table>` ...

- Each element has its own API
 - a. attributes
 - b. properties
 - c. methods
 - d. events

Memory Lane (2)

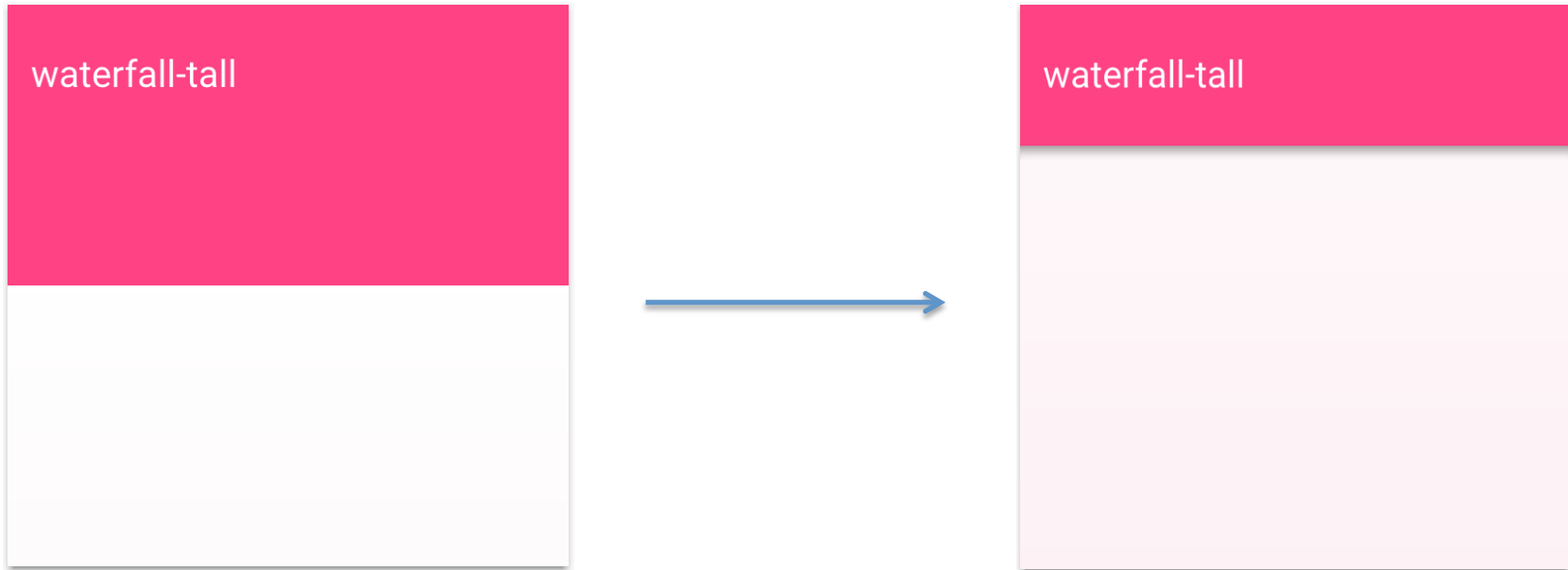
e.g.: To create a fixed header that will be minimize upon scrolling the page



It will take quite a few of HTML mark-up, CSS and JS

Memory Lane (2)

e.g.: To create a fixed header that will be minimize upon scrolling the page



What if, one can create the same widget with:

```
<my-header-panel></my-header-panel>
```


What is Polymer?

- ❖ It is a software library developed by Google
- ❖ It is used to define and style web components i.e. to create custom elements

Objective:

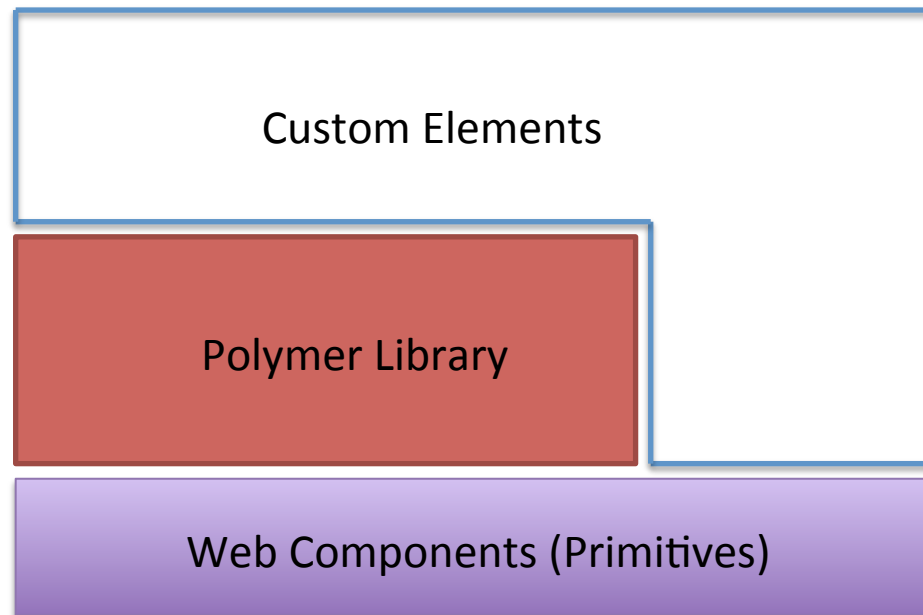
“... designed to make it easier and faster for devolpers to create great, reusable components for the modern web”

Advantage:

easily maintained from both user and developer point of view

What is Polymer? (2)

*built on top of the web components standard and it helps to
build your own custom elements*



What is Polymer? (3)

“Web components are set of standards currently being produced by Google engineers as a W3C specification that allow for the creation of Reusable widgets or components in web documents and web application”

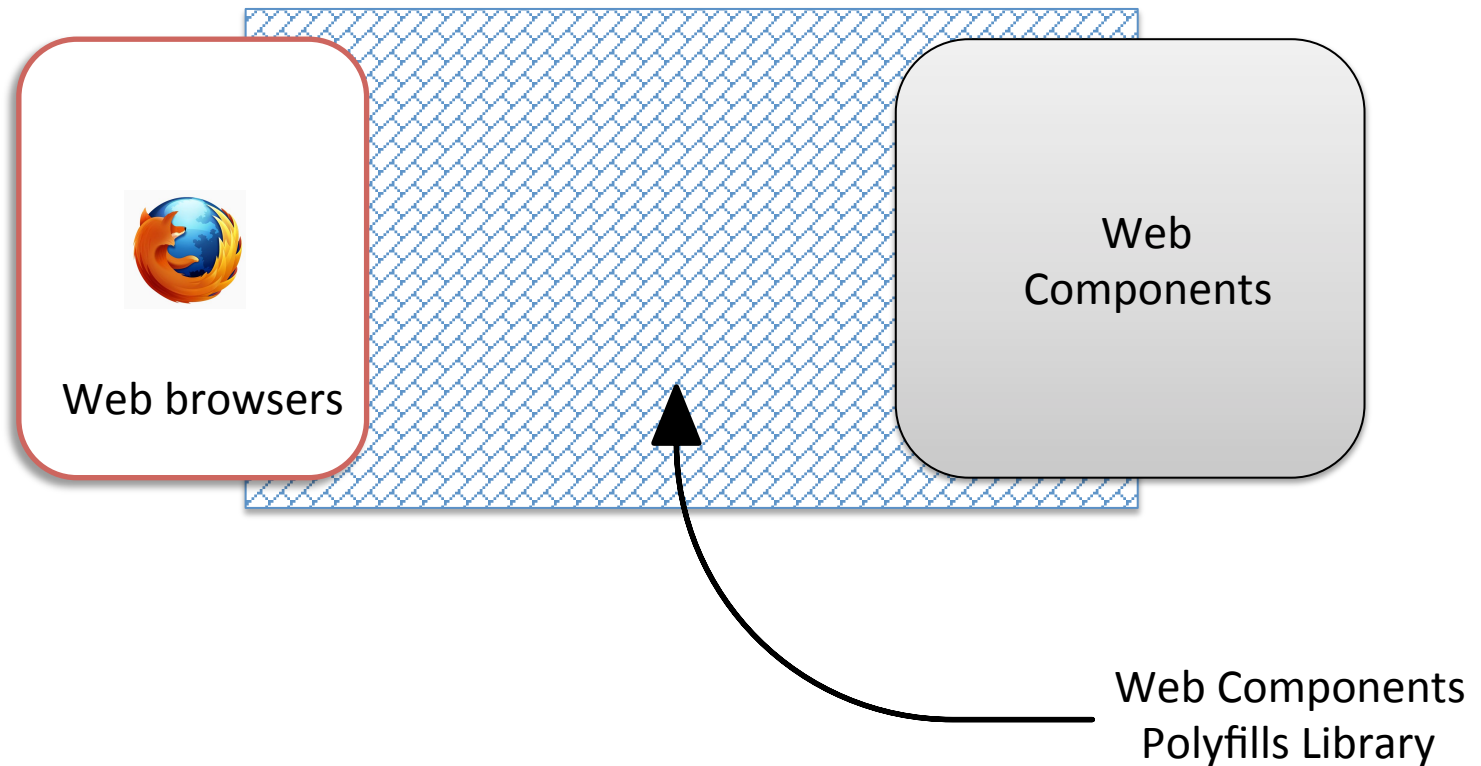
- ◆ In reality you can build your own custom elements using the “primitive” Web Components without POLYMER

the BUT is:

- ☒ it will take lot of work
- ☒ not all browser support W3C standard yet

What is Polymer? (4)

To make polymer available across most browsers



What Polymer offer

✓ Provides a declarative syntax that make it simpler to define custom elements

✓ Provides feature like:

Templating
Two-way binding
Property observation

✓ Provides some built-in custom elements like:

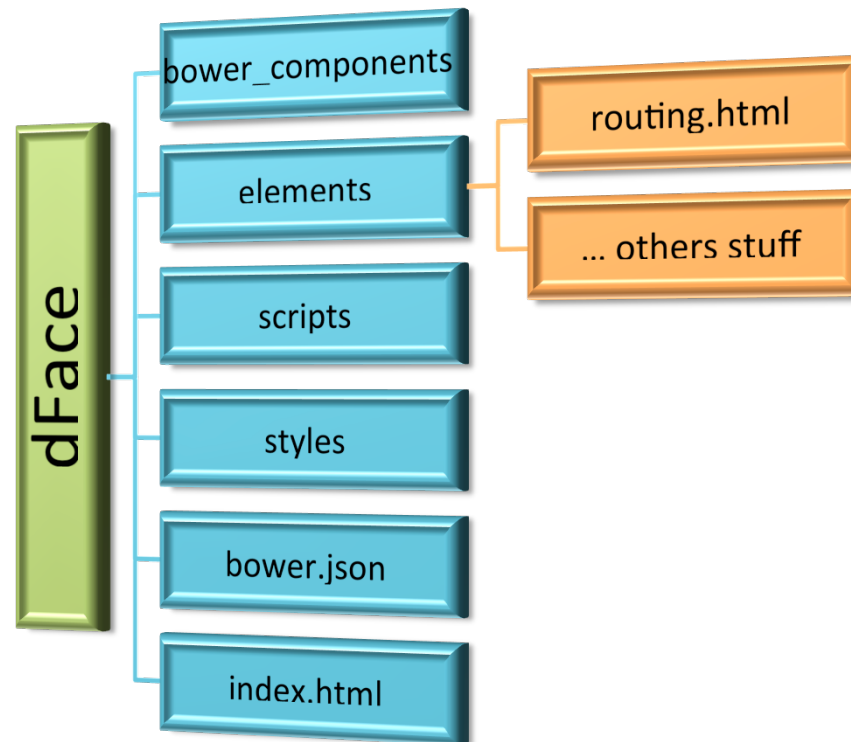
➤ Iron-ajax

➤ Iron-list

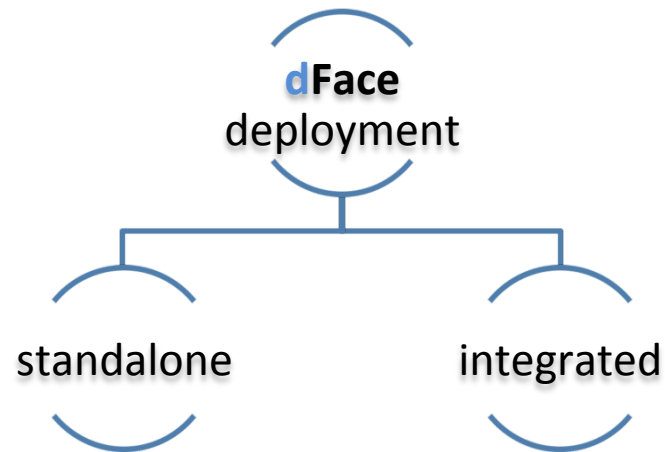
➤ Paper-toolbar

Back to one of the new guys

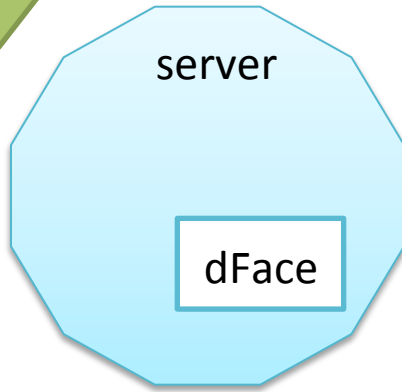
dFace contents



Deployment

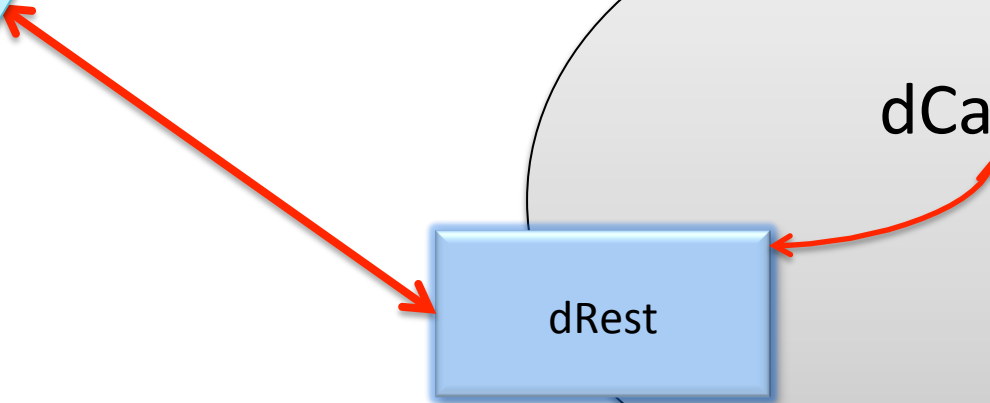
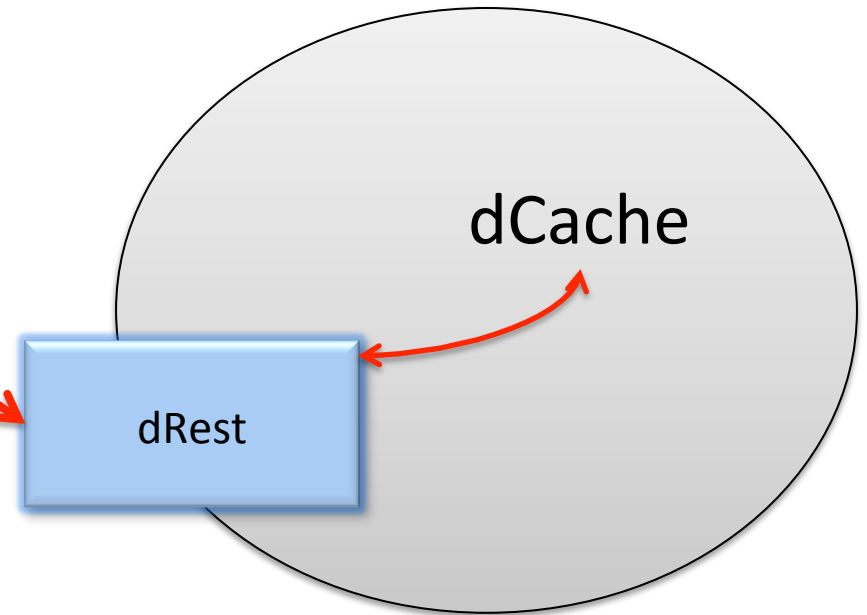
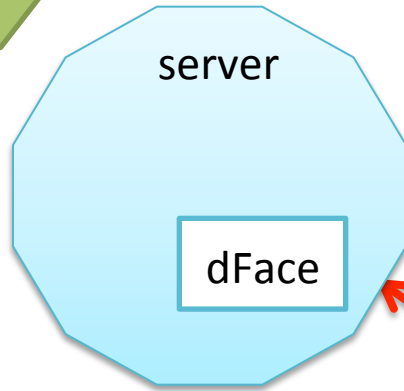


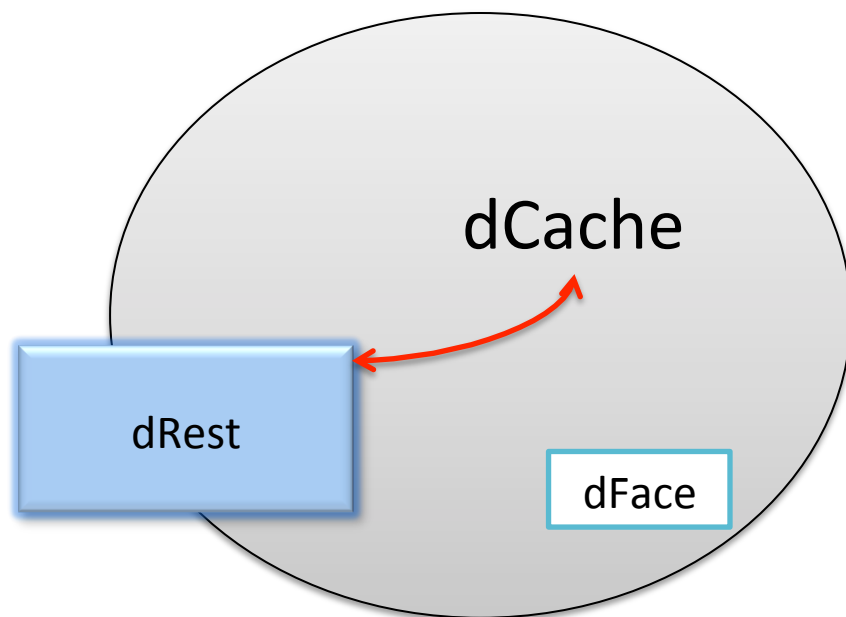
1

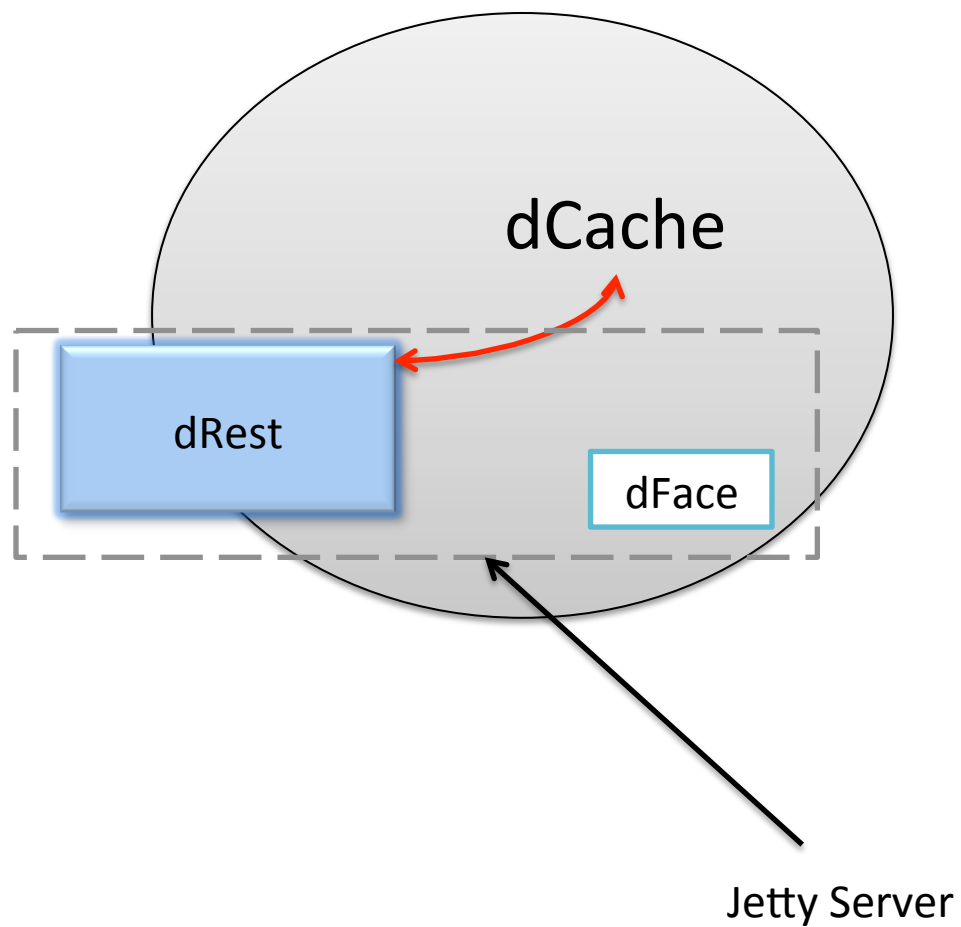


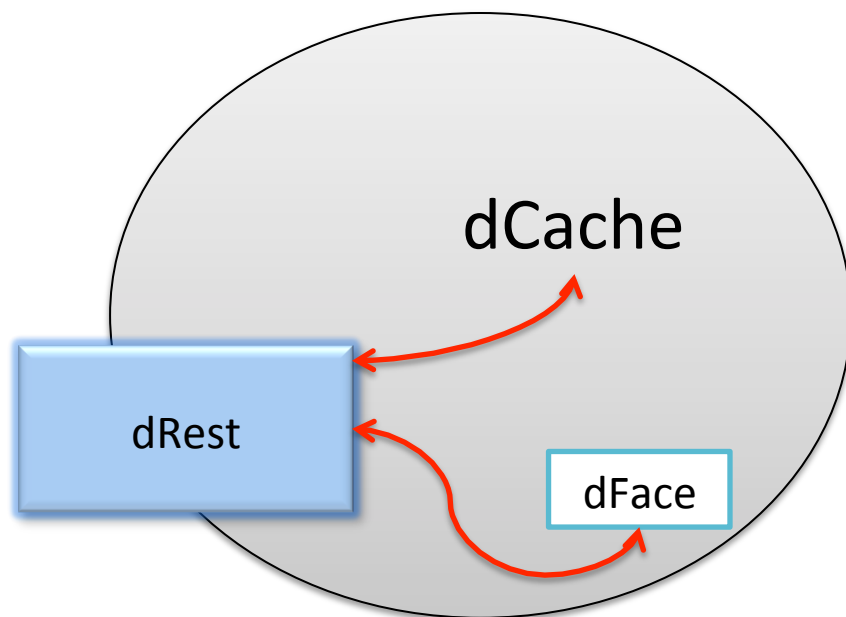


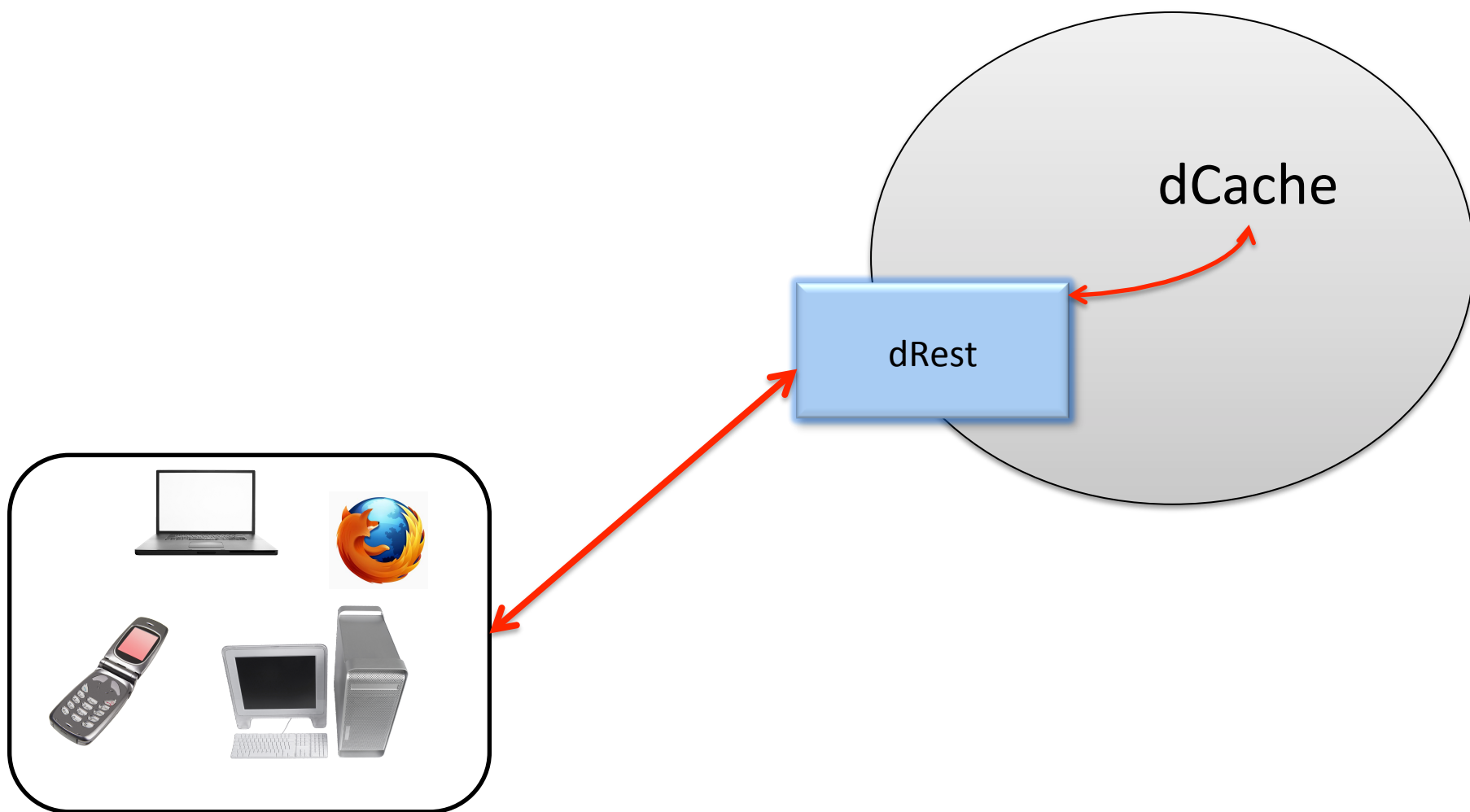
1

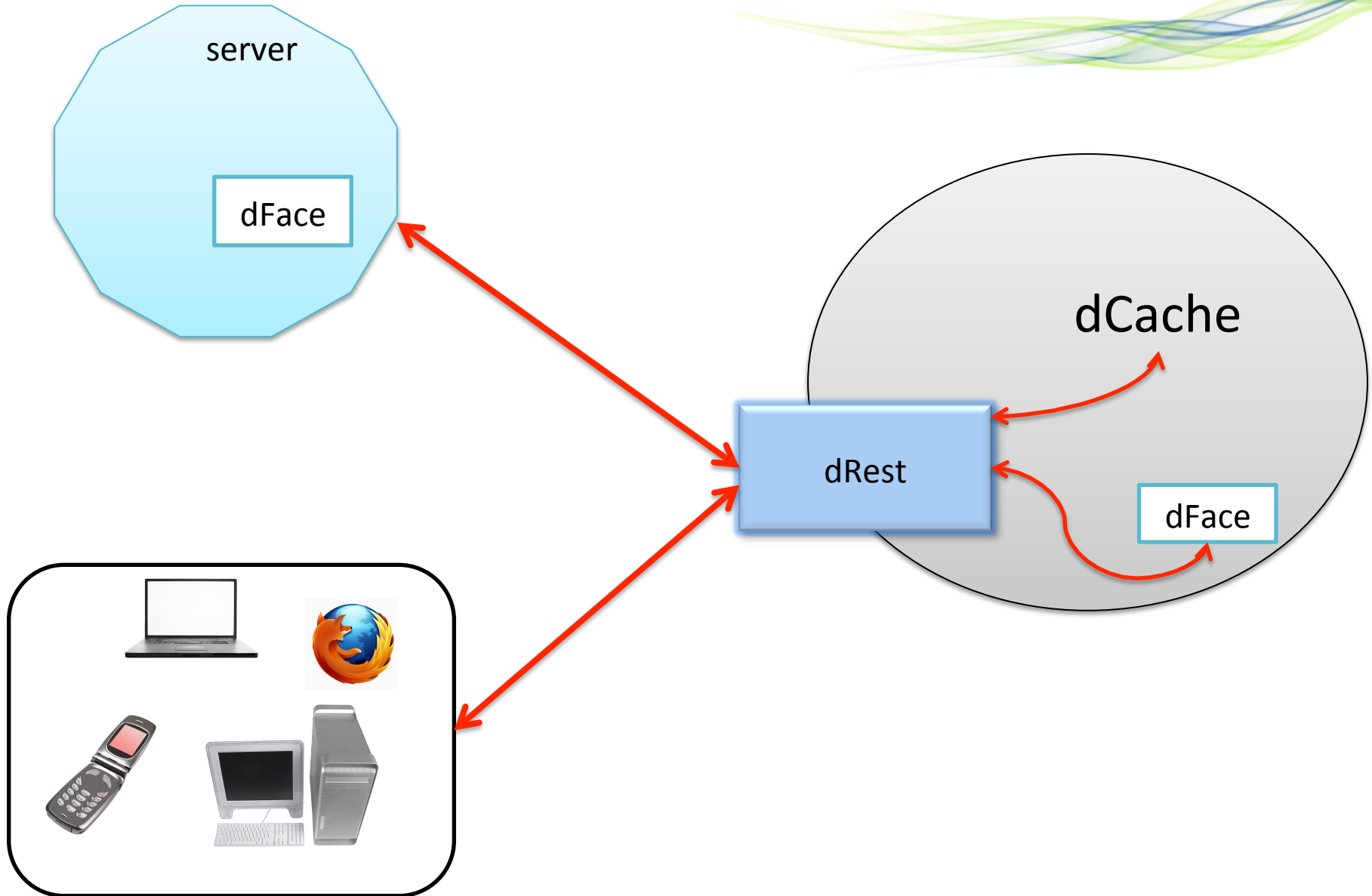












What **dFace** can do at the moment

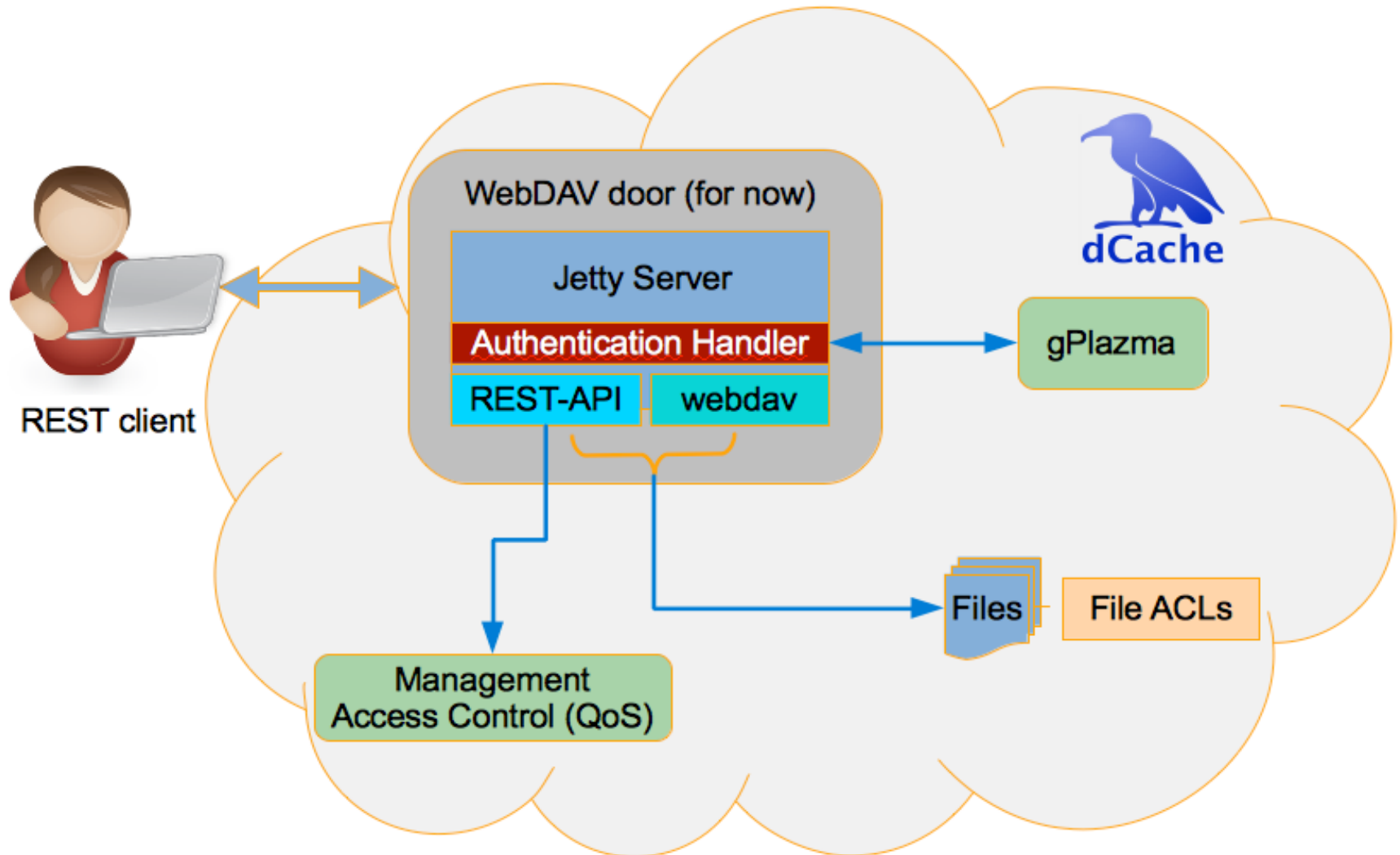
1. Login with username + password
2. Listing of user root files
3. Shows files Metadata
4. Rename a file
5. Download a file
6. Move file(s) into another directory
7. Create a new directory

The second new guy: dRest

Goals of providing RESTful API service

- Provide QoS features and expose them to the user
- Expose **dCache** functionality to end user (e.g. web interface)
 - ✓ Stage, flush, replication, namespace manipulations...
 - ✓ Interact with **dCache** without the need to script against admin interface
 - replacement for chimera-cli
 - ✓ Your software can talk to **dCache**
 - ✓ You can create your own Application

Architecture



List of Directories or Files

```
curl -user username:password  
  -H "Accept: application/json"  
  -H "Content-Type: application/json"  
  -X POST http://localhost:2880/restapi/LIST/  
  --data '{"directoryPath":"/public"}'
```

Result

```
{ "List":  
  [  
    {  
      "size": "378",  
      "creationTime": "Wed Apr 06 15:45:43 CEST 2016",  
      "mtime": "Wed Apr 06 15:45:58 CEST 2016",  
      "fileType": "REGULAR",  
      "storageInfo": "accessLatency=NEARLINE; retentionPolicy=CUSTODIAL,"  
      "fileName": "testFile1",  
      "path": "/public/testFile1"  
    },  
    ...  
  ]  
}
```

File Locality Query

```
curl --user username:password  
  -H "Accept: application/json"  
  -H "Content-Type: application/json"  
  -X GET http://localhost:2880/drestapi/LOCALITY/replica/test00001
```

RESULT:

```
{  
  "Locality": "ONLINE"  
}
```

Demo

Short Term Plans

- House cleaning
- More user friendly interactions like drag n drop
- Login
 - > with certificate, openID connect and any other supported implementation of authenticating & authourisation system
- upload
- Management interface
 - like pin management, staging, etc.

Long Term Plans

- Share and sync
 - Search and Selection
 - recycle bin
 - **User interface for admin**
- and many more ● ● ●

Thank You