Content Personalization in AEM

(By Prabhdeep Singh)

Agenda

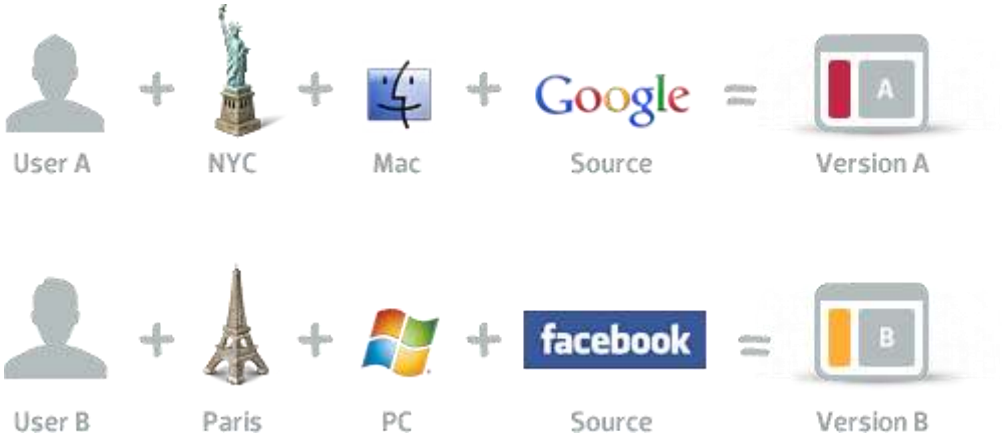
- What is Content Personalization?
- Why do we need Personalization?
- Types of Personalization
- Content Personalization with AEM
- Architecture
- Client Context
- Session Store
- Client Context Manager
- Context Store Components
- Customizing Client Context & Context Store Component
- Demo & Exercise

Personalization

- 500 different people go to Amazon.com and each user sees a different version of the home page.
- How’s that done?
- It's Personalization

Content Personalization

Content Personalization in short means delivering the Right Content to the Right Person at the Right Time.



Why do we need Personalization?

- Personalized emails deliver 6x higher transaction rates.
- Nearly three-fourths (74%) of online consumers get frustrated with Web sites when content (e.g., offers, ads, promotions) appears that has nothing to do with their interests.
- In-house marketers who are personalizing their Web experiences see, on average, a 19% uplift in sales.
- 90% of consumers find custom content useful

Types of Personalization

- **Server Side Personalization**
  - Traditional Method of Personalization
  - Personalization Logic/Rules are defined at server side
  - Very slow performance-heavy process
- **Client Side Personalization**
  - Current Method of Personalization
  - Personalization Logic/Rules are defined at client side and data is fetched from server only when required
  - Fast and better performing process
  - AEM uses this type of personalization

Content Personalization

There are three elements involved in personalization:

- Users/Leads/Visitor
- Content/Experience
- Rules/Segments

Users/Leads/Visitor

- have profiles, both individual and group, containing characteristics which can be used to personalize the content they see.
- take actions that can be analysed and matched against behaviour rules to tailor the content they see.

Content/Experience

- is what the user wants to see to fulfil their tasks.
- can be categorized, and therefore made available to users according to predefined rules
- must, in some way, be dependent upon the user – if every user would see the same content, then personalization would be redundant.

Rules/Segments

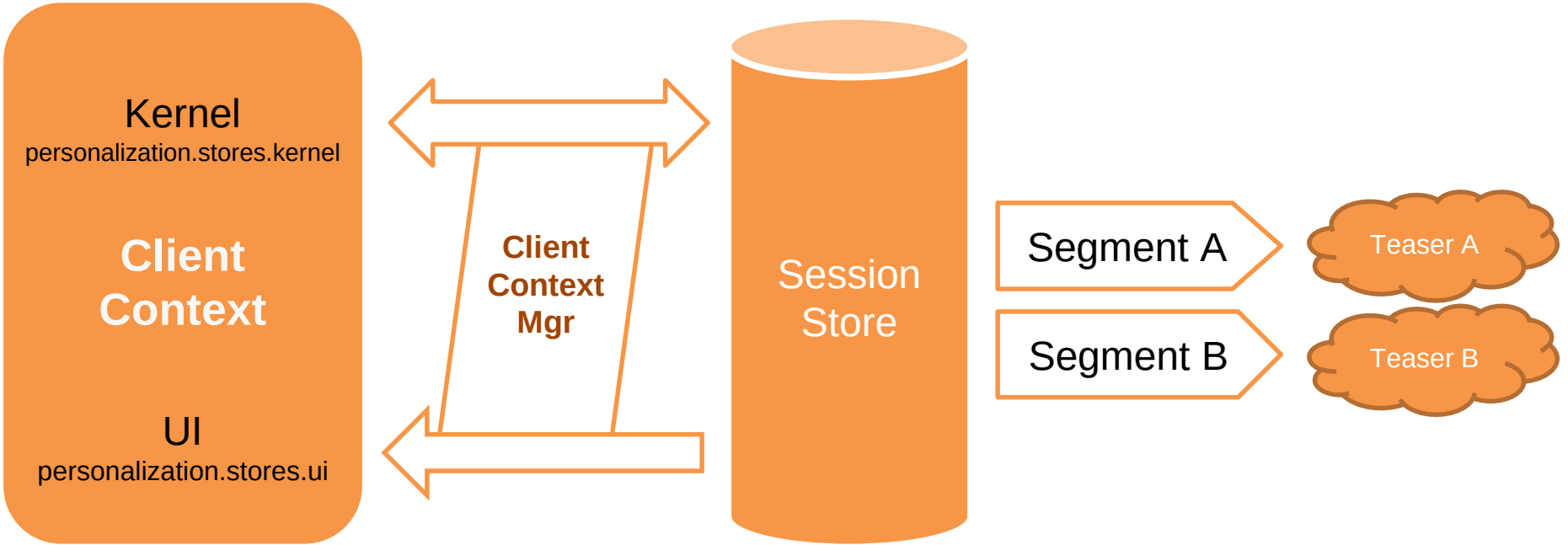
- define how personalization actually happens – which content the user can see, and when.

- also defines which rule gets priority in case of multiple hits.

Lets Start with a Personalization

Demo

Architecture

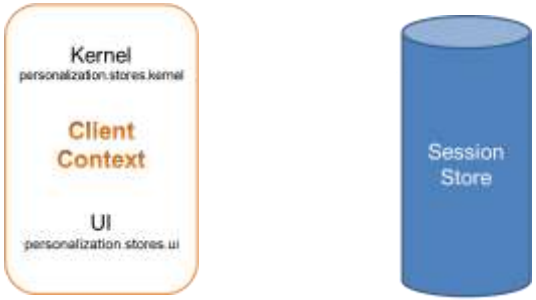


Client Context

- Represents a dynamically assembled collection of user data that can be used to determine exactly what content should be shown on a given web page, in a given situation.
- Provides you with certain information about the current visitor and page.
- Includes various Session Stores that contain user data.
- To see the Client Context, type Ctrl-Alt-C from within any web page.

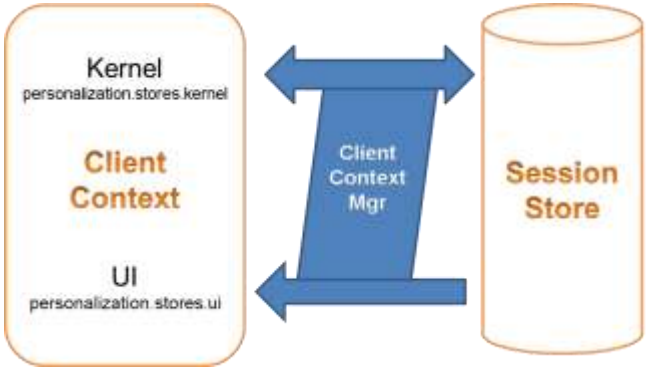
Session Store

- A JavaScript object available both on the author and publish instances.
- Contains the **User Data**.
- A session store can be persisted across browser sessions, or can last only for the browser session in which it is created.
- **Types of Session Stores** *:*
  - CQ_Analytics.SessionStore *(JSONStore and JSONPStore)*
  - CQ_Analytics.PersistedSessionStore *(PersistedJSONStore and PersistedJSONPStore)*



Client Context Manager

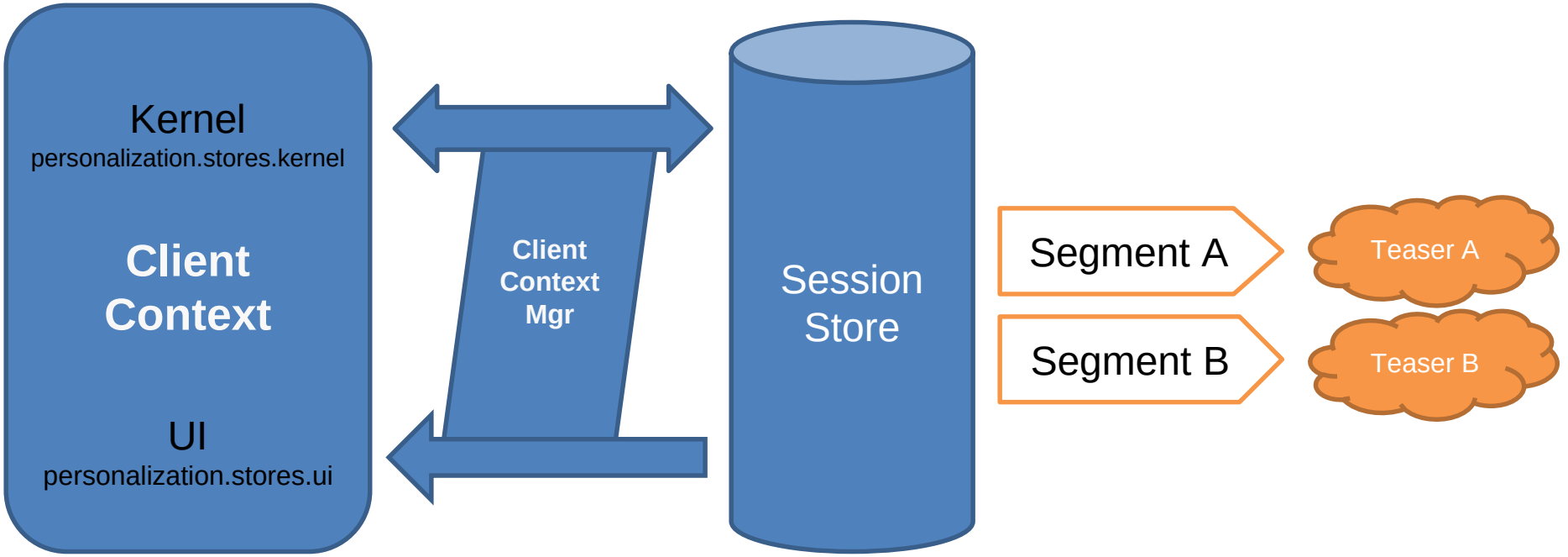
- Acts as an interface to *register*, *persist* and *manage* Session Stores.
- Contains a set of registered Session Stores.
- Is a Singleton object.
- Extends CQ_Analytics.PersistedSessionStore.



Context Store Components

- Displays data from a session store with which they are associated.
- A context store component is a CQ component that can be added to Client Context

Architecture



Custom Session Store Creation Steps

TO THE NEW DIGITAL

1. Include Client Context to a page.
2. Create two Client Libraries – Kernel & UI.
3. Kernel
   - Register a New Session Store.
   - Add a method to load data into the Session Store, from a servlet or directly an external source.
4. UI
   - Create a Context Store Component and add to Client Context.
   - Add personalization:storeRendererTag in Component's JSP to call 'renderer()' Method.
   - Define renderer() method in JS file.
5. Automation: Handle the synchronization via registering 'update', 'initialize' and 'change' events.
6. Initialize the session store in the 'init.js.jsp'.

Include Client Context To A Page

- Include the Client Context component to the body section of your web pages to enable Client Context.

  ```
  <cq:include path="clientcontext" resourceType="cq/personalization/components/clientcontext"/>
  ```

- The path of the Client Context component node is:

  ```
  /libs/cq/personalization/components/clientcontext
  ```

- To see the Client Context in operation, type Ctrl-Alt-C from within any web page.

Create Two Client Libraries

- As Context Store component is only used by author for simulation, we don’t want it to be loaded on the publish instance.
- Thus, we create two JavaScript libraries with the following categories:
    - Kernel : personalization.stores.kernel
    - UI : personalization.stores.ui

Register a New Session Store

Create a JS file in the kernel clientlibs and register a new instance of Custom Session Store

```
CQ_Analytics.CustomStoreMgr =
CQ_Analytics.JSONStore.registerNewInstance(“custom_name")
```

OR

```
CQ_Analytics.CustomStoreMgr =
CQ_Analytics.PersistedJSONStore.registerNewInstance(“custom_name")
```

Load Data Into Session Store

- Create a **servlet** and which returns data in a JSON object.
- Add a loadData() method in JS File to load the JSON data from the servlet and store the JSON data in the ‘data’ property of Store Manager Object.

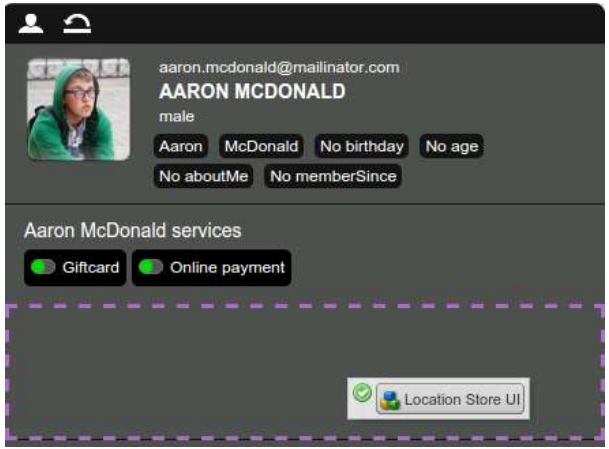
CQ_Analytics.CustomStoreMgr.data = CQ.shared.HTTP.eval(“URL”)

- Lets see the data loaded in the session store in console:

CQ_Analytics.CustomStoreMgr.loadData()

Create a Context Store Component and add to Client Context

- We will now display the data on the client context by creating a custom Client Context Component.
- Create a new CQ component with the properties :
  - sling:resourceSuperType = cq/personalization/components/contextstores/genericstoreproperties
  - componentGroup = Client Context
- Drop this component in the Client Context by editing it.



Add Rendering logic to Client Context component

- Add <personalization:storeRendererTag> to the component's JSP, which calls *renderStore* method which in-turn calls the *renderer* method on the session store and register a method on the update event

```
<%@taglib prefix="personalization" uri="http://www.day.com/taglibs/cq/personalization/1.0" %>
<div class="cq-cc-store">
        <personalization:storeRendererTag store="customstore"/>
</div>
```

- Create a renderer method in the UI JS and write the required Logic in it.

Loading & Updating Store Data Automatically

- Now, handle the synchronization, i.e. store should be loaded by itself if it is present on client context page and updates if and when required.
- This is done by registering following events :
  - ‘update’ event on Profile: `CQ_Analytics.ProfileDataMgr.addListener("update", function() { … });`
  - ‘initialize’ event on Session store: `CQ_Analytics.CustomStoreMgr.addListener("initialize", function() { … })`
  - ‘change’ event on Input(Checkbox): `$CQ(".customstore-input").change()`

Initialize Session store for the First Time

- Create a file init.js.jsp associated with the Context Store component.
- This file generates the Javascript code that initializes the session store that our context store component uses.
- Remember : The code must set mime type of the file to “text/javascript”

```
<%@ page session="false" contentType="text/javascript" %>
if (CQ_Analytics.CustomStoreMgr) {
        CQ_Analytics.CustomStoreMgr.init();
}
```

Before moving ahead, time for some

Segmentation Revision..

Segmentation Revision

- **Brand** : Highest Level Entity
  - Example :- ABC Software is a brand
  - Consists of campaigns
- **Campaigns** : A way in which a brand is endorsed.
  - Example :- ABC firm running campaigns to sell its services
  - Consists of experiences
- **Experiences** : It is what is visible to the user
  - Could be a teaser page to make user use a service or a newsletter sent to intimate user of some kind of new service
- **Segment :** Used to segregate users based on some criteria.
  - Example :- segment of users with age more than 30 and living in India
  - http://localhost:4502/miscadmin#/etc/segmentation

Segmentation Revision

- **Boost Factor :** Used to give priority to some of the user segments.
  - There could be multiple segments a user can belong to. A segment with more 'Boost Factor' is given preference over the one with lower 'Boost Factor‘
- **Clickstream Score**, is based on the tags and related tag hits held within the visitor's client context (show how often a visitor has clicked on pages containing the respective tag). The hit rates for tags defined on the teaser page are compared.

Finally, Use the data for Personalization

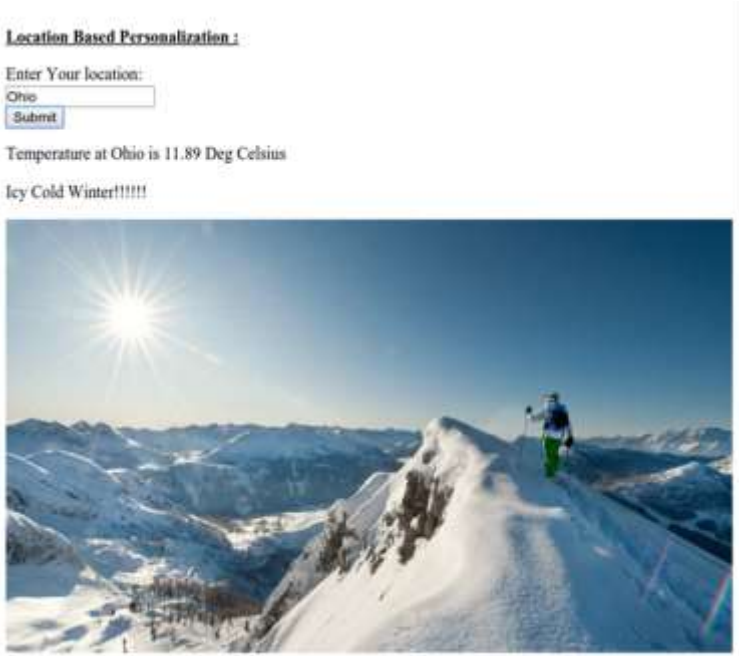
- The last step is to personalize content based on the value of the data by using a segment based on the value of the property of the data store.
- “Generic Store Property” component is used to segment users based on Session Store’s Data Properties.

Prolog.jsp & Epilog.jsp

- Component interaction that allows you to add server side logic before or after the component rendering in case of prolog.jsp and epilog.jsp respectively.
- Couldn’t find any Use-case scenario!!! Lets Brainstorm...

Exercise

Taking the ‘Location Name’ as an input from the visitor, ‘Current Location’ set as default, find out the temperature in that Location using Open Weather API and show different teasers for Summer and Winter seasons, where Temperature > 20 Degree Celsius is taken as Summer, otherwise winter. Also, refreshing the page should retain the data stored in the Session Store.



Open Weather URL with my User Id :

1) **For LocName** :
   http://api.openweathermap.org/data/2.5/weather?q=‘location'&units=metric&appid=519dd4cfe51d4e65be72974fbea0ae3e
2) **For Latitude & Longitude** :
   http://api.openweathermap.org/data/2.5/weather?lat={Lat}&lon={lon}&units=metric&appid=519dd4cfe51d4e65be72974fbea0ae3e


THANK
YOU
GRACIAS
ARIGATO
SHUKURIA
JUSPAXAR
DANKSCHEEN
BOLZÏN
MERCI
BÏYAN
SHUKRIA
TINGKI
YAQHANYELAY
TASHAKKUR ATU
SUKSAMA
EKHMET
GRAZIE
MEHRBANI
MAAKE
KOMAPSUMNIDA
GOZAIMASHITA
EFCHARISTO
SPASIBO
DENKUJA
SPASSIBO
SNACHALHUYA
NUHUN
CHALTU
WABEEJA
MAITEKA
YUSPAGARÀTAM
HUI
GUI
HATUR
EXOJU
SIKOMO
MAKETAU
MINMONCHAR
PALDIES
LAH
ATTO
ANIMA
DHANYABAAD
UNALCHÉESH
MERSI
SAIKO
MERASTAWHY
GAEJTHO
TAVTAPUCH
MEDAWAGSE
AGUY.JE
FAKAAUE
BAIKA