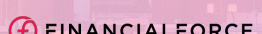
Welcome to

dreamOlé©



#dreamOle18

Barcelona 2018

Using & Building Lightning Service Components

Philippe Ozil
Senior Developer Evangelist, Salesforce
@PhilippeOzil

Forward-Looking Statement

Statement under the Private Securities Litigation Reform Act of 1995

This presentation may contain forward-looking statements that involve risks, uncertainties, and assumptions. If any such uncertainties materialize or if any of the assumptions proves incorrect, the results of salesforce.com, inc. could differ materially from the results expressed or implied by the forward-looking statements we make. All statements other than statements of historical fact could be deemed forward-looking, including any projections of product or service availability, subscriber growth, earnings, revenues, or other financial items and any statements regarding strategies or plans of management for future operations, statements of belief, any statements concerning new, planned, or upgraded services or technology developments and customer contracts or use of our services.

The risks and uncertainties referred to above include – but are not limited to – risks associated with developing and delivering new functionality for our service, new products and services, our new business model, our past operating losses, possible fluctuations in our operating results and rate of growth, interruptions or delays in our Web hosting, breach of our security measures, the outcome of any litigation, risks associated with completed and any possible mergers and acquisitions, the immature market in which we operate, our relatively limited operating history, our ability to expand, retain, and motivate our employees and manage our growth, new releases of our service and successful customer deployment, our limited history reselling non-salesforce.com products, and utilization and selling to larger enterprise customers. Further information on potential factors that could affect the financial results of salesforce.com, inc. is included in our annual report on Form 10-K for the most recent fiscal year and in our quarterly report on Form 10-Q for the most recent fiscal quarter. These documents and others containing important disclosures are available on the SEC Filings section of the Investor Information section of our Web site.

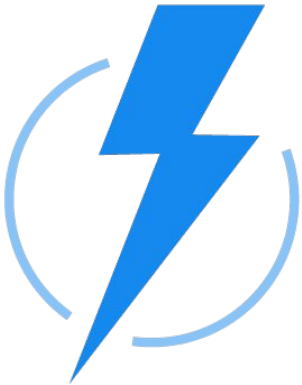
Any unreleased services or features referenced in this or other presentations, press releases or public statements are not currently available and may not be delivered on time or at all. Customers who purchase our services should make the purchase decisions based upon features that are currently available. Salesforce.com, inc. assumes no obligation and does not intend to update these forward-looking statements.

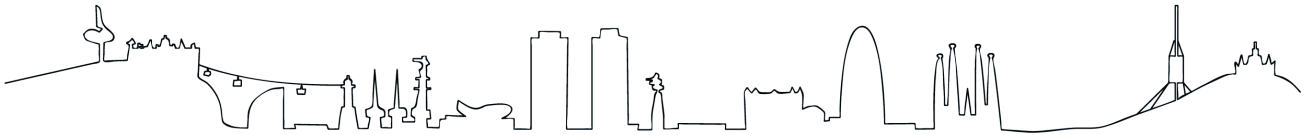
Agenda

Lightning service components

Base service components

How to build your own





Lightning Service Components

Provides services (API)

Generic & reusable

No graphical representation

Using Base Service Components

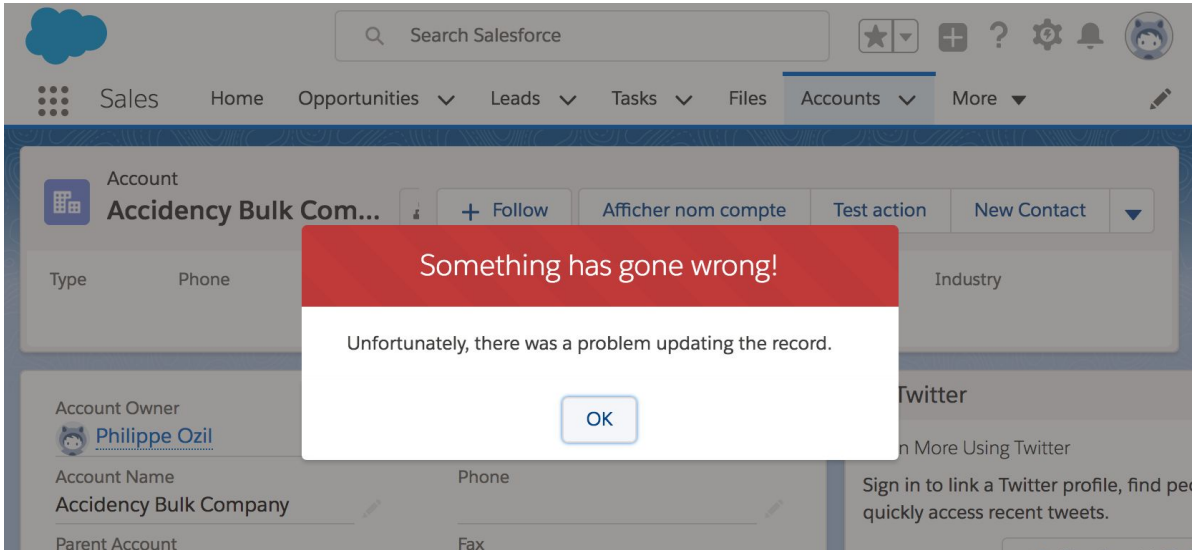
Base Service Components

As of Spring ’18:

- Lightning Data Service
- Notification library
- Overlay library
- Workspace API
- Omni-Channel Toolkit API (Beta)
- Utility Bar API

and more to come…

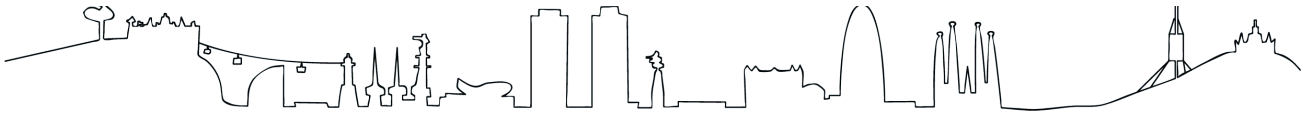
Example: Notification Library



Example: Notification Library

```
<aura:component>
    <lightning:notificationsLibrary aura:id="notifLib"/>
    <lightning:button name="notice" label="Show Notice" onclick="{!c.handleShowNotice}"/>
</aura:component>
```

```
handleShowNotice : function(component, event, helper) {

}
```



Building a Custom Service Component

Identifying a Custom Service Candidate

Can it be reused?

- simplifies component's code
- avoids code duplication

Calling a server-side action is a great match!

+ tens of line of code

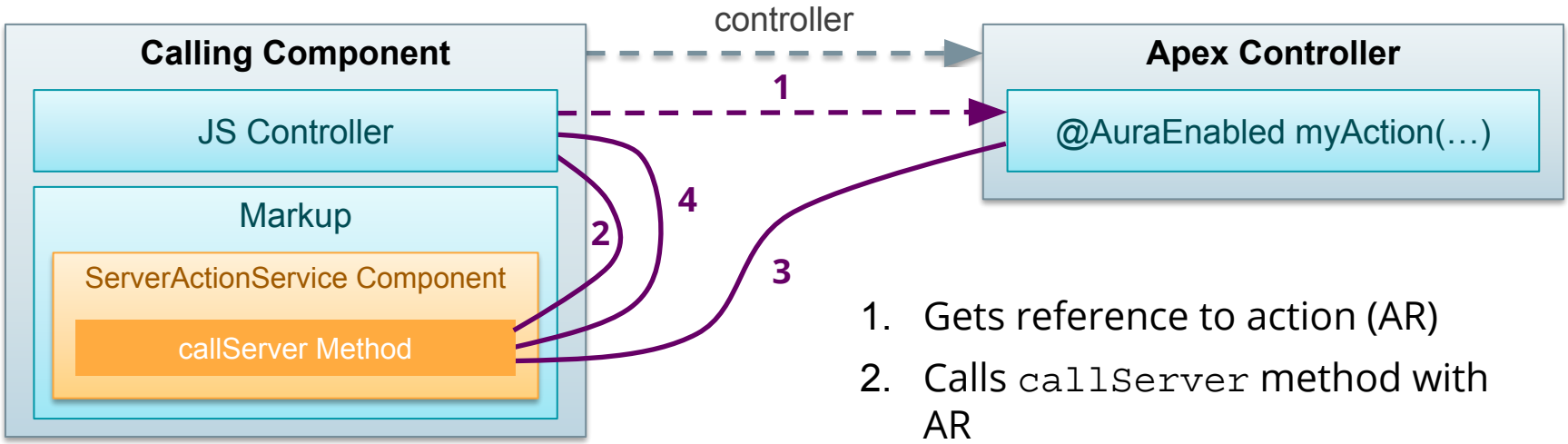
```
myServerSideAction : function(cmp, someParams) {
    var action = cmp.get("c.myAction");

    action.setParams(someParams);

    action.setCallback(this, function(response) {
        var state = response.getState();
        if (state === "SUCCESS") {
            // Do something with response
        }
        else if (state === "ERROR") {
            var errors = response.getError();
            if (errors) {
                // Handle errors
            } else {
                // Handle unknown error
            }
        }
    });

    $A.enqueueAction(action);
}
```

Introducing the **Server Action Service**
(~100 lines of JS)

Server Action Service

Architecture

https://github.com/pozil/server-action-service



1. Gets reference to action (AR)
2. Calls `callServer` method with AR
3. Executes action and gets response
4. Calls success or error callbacks + does error notification

Server Action Service

https://github.com/pozil/server-action-service

Using the service (<10 lines of code)

```
<aura:component controller="MyApexController">

    ...
</aura:component>

callMyServerSideAction : function(cmp, someParams) {

}
```

Architecture Bad & Best Practices (1/2)

When building your service…

⊗ Avoid inheritance

```
<aura:component extends="c:myParentComponent">
    ...
</aura:component>
```

✓ Prefer composition

```
<aura:component>
    <c:myServiceComponent aura:id="myService"/>

    ...
</aura:component>
```

Architecture Bad & Best Practices (2/2)

Expose services via...

⊗ Events

```
<aura:handler event="c:myServiceRequestEvent" action="{!c.doSomething}"/>
<aura:registerEvent name="myServiceNotificationEvent" type="c:myServiceNotificationEvent"/>
```

✓ Methods + optional callback functions

```
<aura:method name="doSomething" action="{!c.doSomething}">
    <aura:attribute name="params" type="Map"/>
    <aura:attribute name="successCallback" type="Function"/>
</aura:method>
```

Closing words

Overview of Lightning Service Components

Efficient and safe way to reuse code

Use them and build your own!

Q & A


copa.do
omega crm
Cognizant
FORMTITAN
resco.net
technology on the move
MAKEMECLOUD
consulting
aircall
blue-infinity
Linked by isobar
Dentsu Aegis Network
FINANCIALFORCE
wefox
BLACKBIRD