salesforce success services

Visualforce to Lightning

Problem Statement (Resolved)

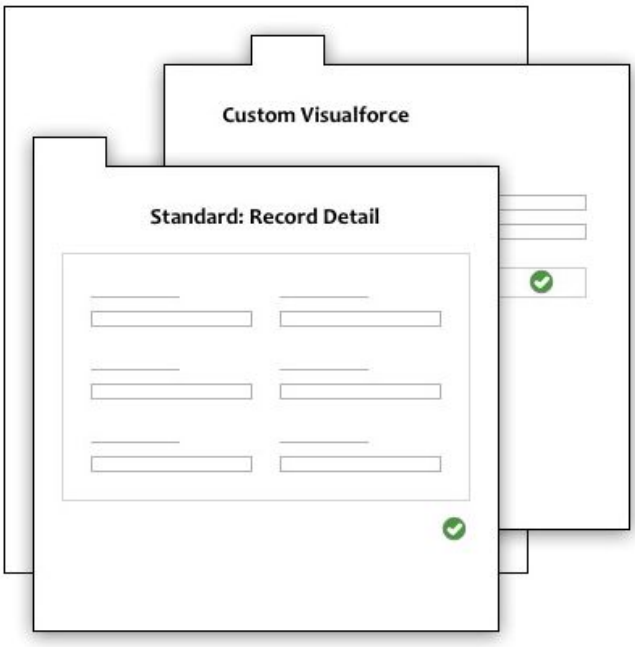
As a Lightning Experience end user, when I make a change to a field in Visualforce, I want to see that change reflected in the Record Detail or the Highlights Panel.

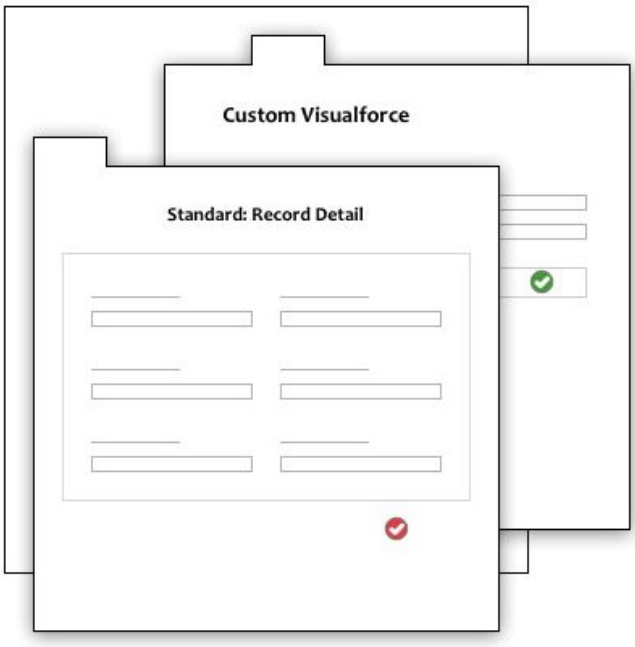
In the inverse of the other issue, when I make a change in Visualforce
There isn't a standard means to update the values in the rest of the Lightning Experience.
(it still shows the old value / Inconsistency)

This can cause confusion and loss of confidence, as the end user isn't sure if the change was actually made.

Or the user has to refresh the page each time a change was made, which causes more than a little frustration.

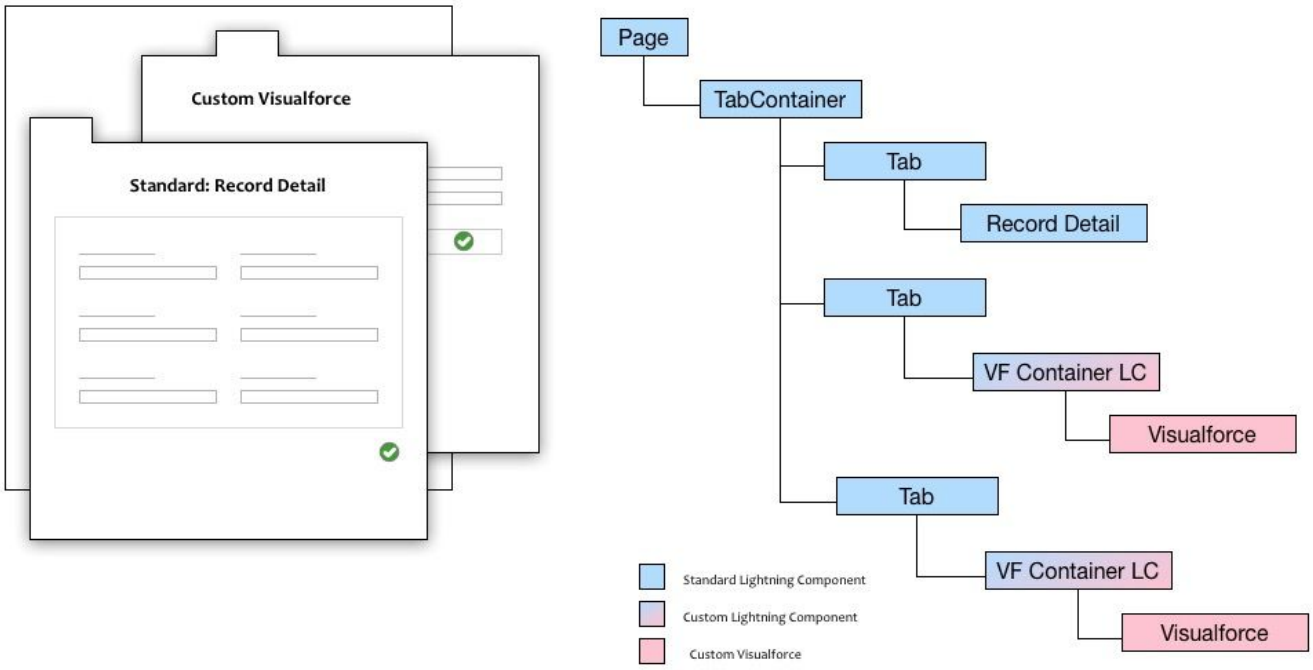
Video demonstration

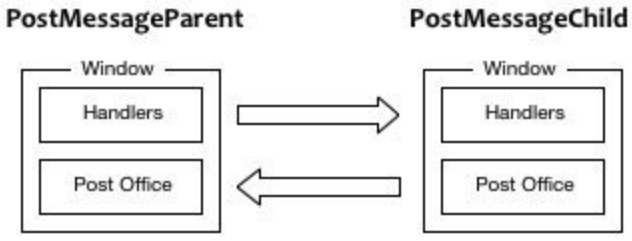




Solution Overview

To allow Visualforce to Communicate with standard Lightning Experience components,
We utilize a custom Lightning Component to contain the visualforce pages
And utilize PostMessages[1] to communicate between the two.





[1] https://developer.mozilla.org/en-US/docs/Web/API/Window/postMessage

PostMessages are meant to provide an event driven means of communicating between different domains (i.e. lightning.salesforce.com and visualforce.salesforce.com – the firewall between the two)

Dispatching Lightning Events from Visualforce

The simplest thing to cover is dispatching Lightning Events from Visualforce
(Visualforce to Visualforce communication just adds on top)

When the page initially loads, the Visualforce Container (custom Lightning Component) asks the window to addEventListener for messages.

(Because Lightning communicates through PostMessages, our custom messages will be filtered to only those from visualforce.salesforce.com, but additional monitors can be added)

When the page needs to dispatch an event (such as the force:refreshView event – to update the standard Highlights and Record Detail), the custom Visualforce page just sends a postMessage to the parent window – that the Visualforce Container will receive.

That message can be parsed, and once validated, a new Lightning Event can be dispatched
So that that it can ultimately reach the Hightlights panel and Record Details.

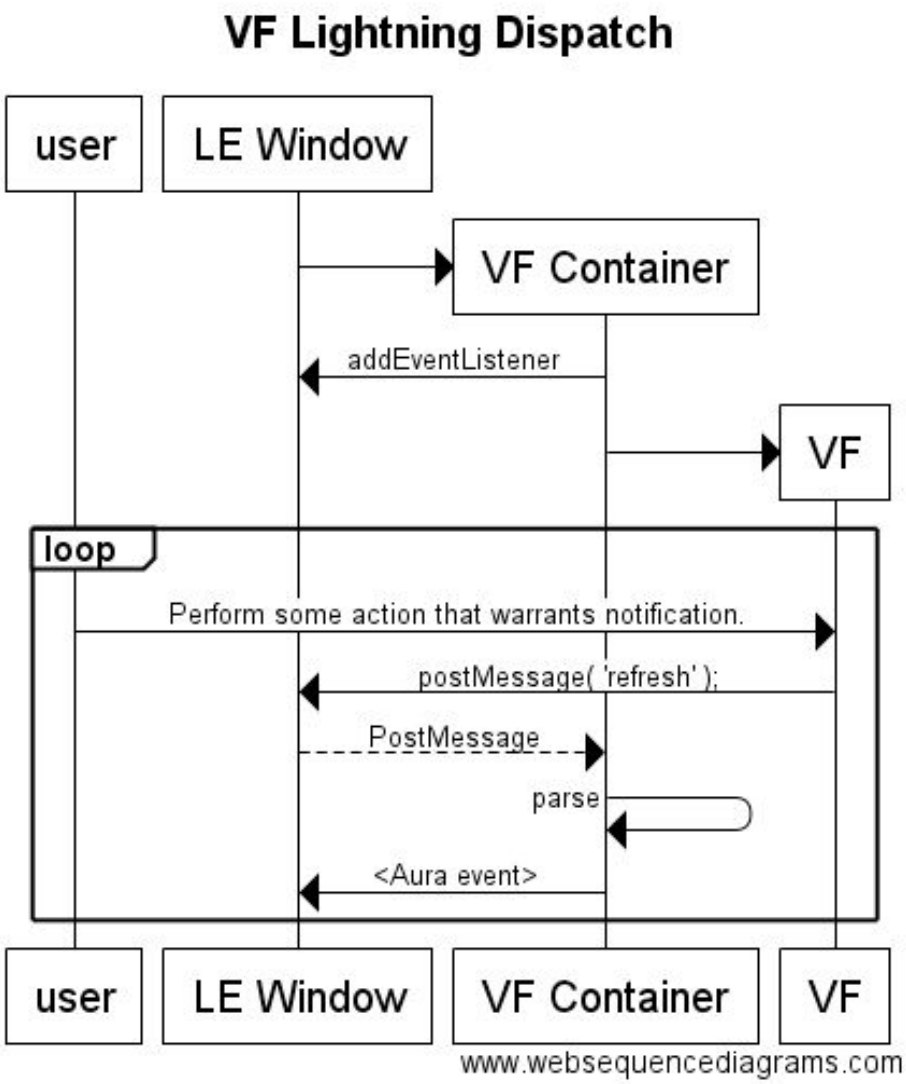
(Diagram below)

Note: it is recommended that the Visualforce Container perform some validation to ensure that the postMessage is only handled once.

Unfortunately postMessages are immutable strings, so they cannot be changed (like adding on handled=True) so others can ignore it, or stop propogation so others don't hear it.

The example provided provides the GlobalId of the component (passed by URL) so only that component will handle it.

It might be possible to dispatch an Application level event to let other Containers know to ignore it, but this appears to be a race scenario (the others would be told to ignore after they hear it).



[2]

[2] If anyone wants to update / review – https://www.websequencediagrams.com/?lz=dGl0bGUgVkYgTGlnaHRuaW5nIERpc3BhdGNoCgpwYXJ0aWNpcGFudCB1c2VyAAQNTEUgV2luZG93ABoNVkYgQ29udGFpbgAgD1ZGCgoKACsJLT4qAB4MOgoALAwgLT4AVAo6IGFkZEV2ZW50TGlzdGVuZXIAGxEqVkY6Cgpsb29wIAp1cwA6BlZGOiBQZXJmb3JtIHNvbWUgYWN0aW9uIHRoYXQgd2FycmFudHMgbm90aWZpY2F0aW9uLgpWRgBrD3Bvc3RNZXNzYWdlKCAncmVmcmVzaCcgKTsAgT8KIC0AaAUAgTILOiBQAC0KAIFHEQCBaQ0gcGFycwASEgCBcQs8QXVyYSBldmVudD4&s=default

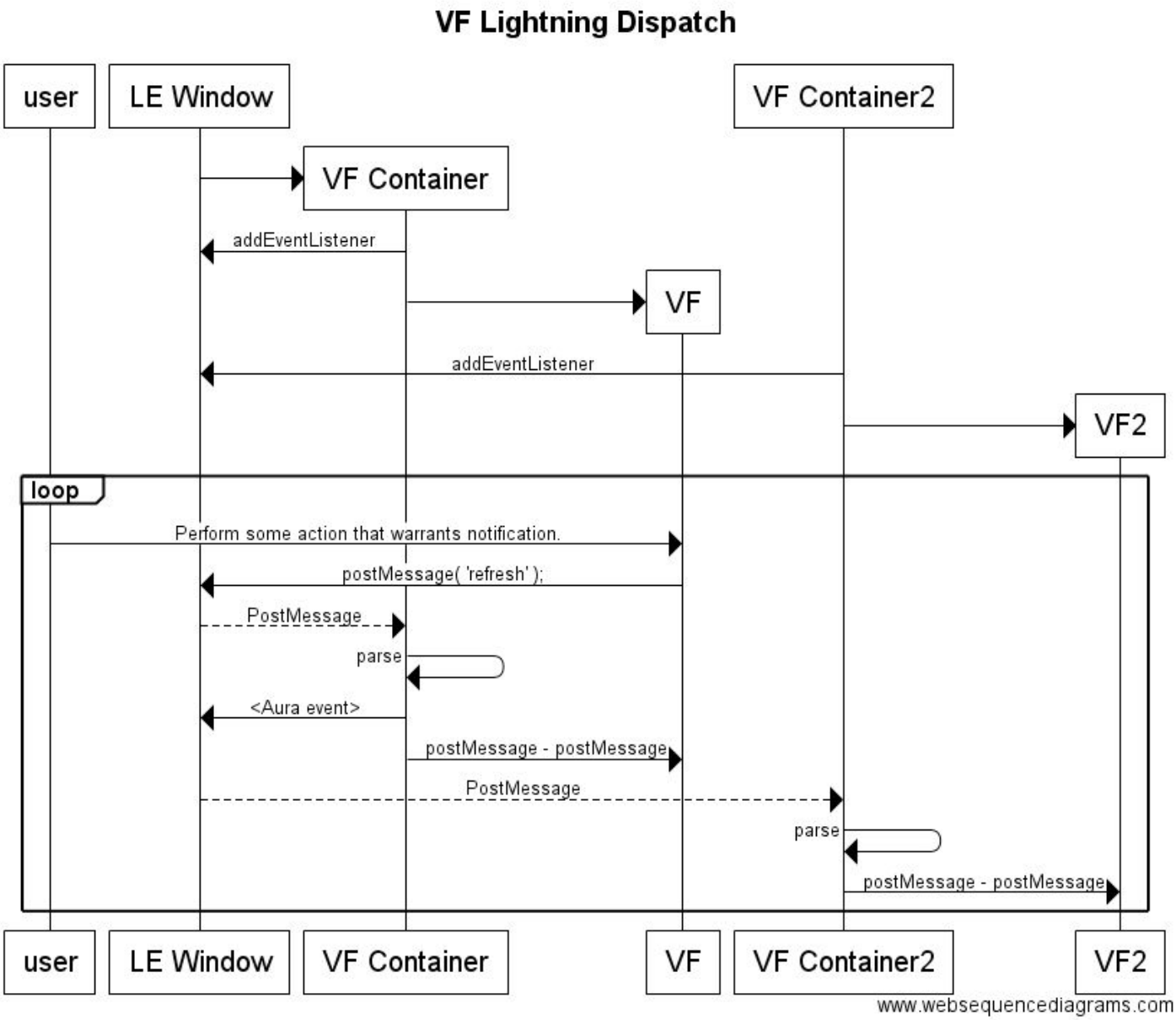
Visualforce to Visualforce Updates

This simply extends on the previous example, but this time, the PostMessage is monitored by more than one Visualforce Container.

In this case, each Visualforce Container simply hears the event, but can dispatch the the event to only the contained iframe.

This can be done by adding on an aura:id to the contained iFrame, and dispatching through something like the following:

```
component.find( "targetFrame").getElement().contentWindow.postMessage(...)
```



3

[3] If anyone wants to edit / review: https://www.websequencediagrams.com/?lz=dGl0bGUgVkYgTGlnaHRuaW5nIERpc3BhdGNoCgpwYXJ0aWNpcGFudCB1c2VyAAQNTEUgV2luZG93ABoNVkYgQ29udGFpbgAgD1ZGAA8ZMgAzDzIKCgoAVQktPioASAw6CgBWDCAtPgB-CjogYWRkRXZlbnRMaXN0ZW5lcgAbESpWRgAzDjIAFiwAKAUqVkYyOgoKbG9vcCAKdXMAfgZWRjogUGVyZm9ybSBzb21lIGFjdGlvbiB0aGF0IHdhcnJhbnRzIG5vdGlmaWNhdGlvbi4KVkYAgS8PcG9zdE1lc3NhZ2UoICdyZWZyZXNoJyApOwCCAwogLQBoBQCBdgs6IFAALQoAggsRAIItDSBwYXJzABISAII1CzxBdXJhIGV2ZW50PgA5EwCBFg0gLQCBJQwAgyYLAIETETIAgQobAIJ7BQCDfw0AgRMUAB8HMgBmGw&s=default

Example Code

Please see the attached example code.

It can be found in Github here:

https://github.com/paulroth3d/VisualforceToLightning

Helpers

We also made a couple quite interesting classes to help along the way.

Specifically:

- LNE_PostMessage2
- LNE_MessagePostOffice

(but I also included a very nice little helper for working with remoting from visualforce – used in the example and a couple others.)

https://git.soma.salesforce.com/proth/VisualforceToLightning

We ended up moving away from standard ES6 Classes due to some unfortunate bugs with current Chrome in handling multiple declarations of the same class.

Any thoughts are welcome