

San Diego Java Users Group

April 16, 2019

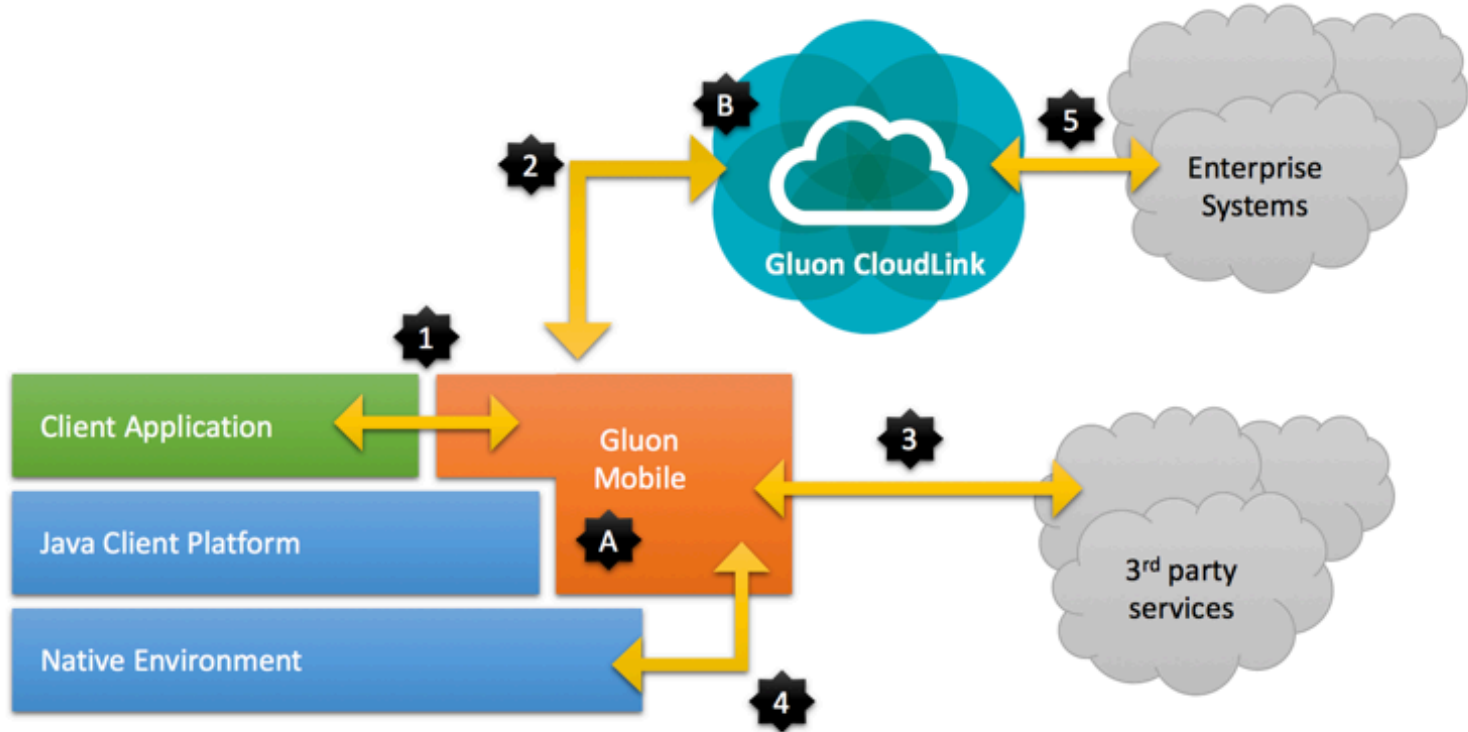
JavaFX Now and Beyond Desktop to Mobile

Paul Anderson
Gail Anderson
Anderson Software Group, Inc.
asgteach.com

Agenda

- ▶ Why JavaFX on Mobile?
- ▶ Gluon Framework
- ▶ Afterburner Framework
- ▶ Gluon Connect and Gluon CloudLink
- ▶ Oracle Cloud Database
- ▶ Oracle REST Data Services
- ▶ Wrap Up, Q & A

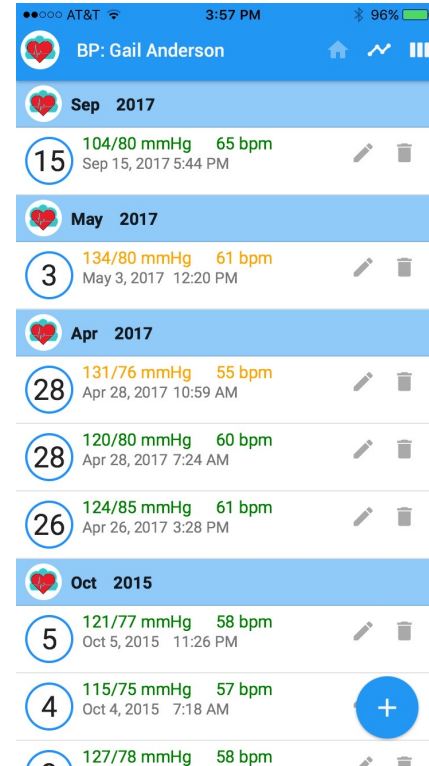
Gluon Framework



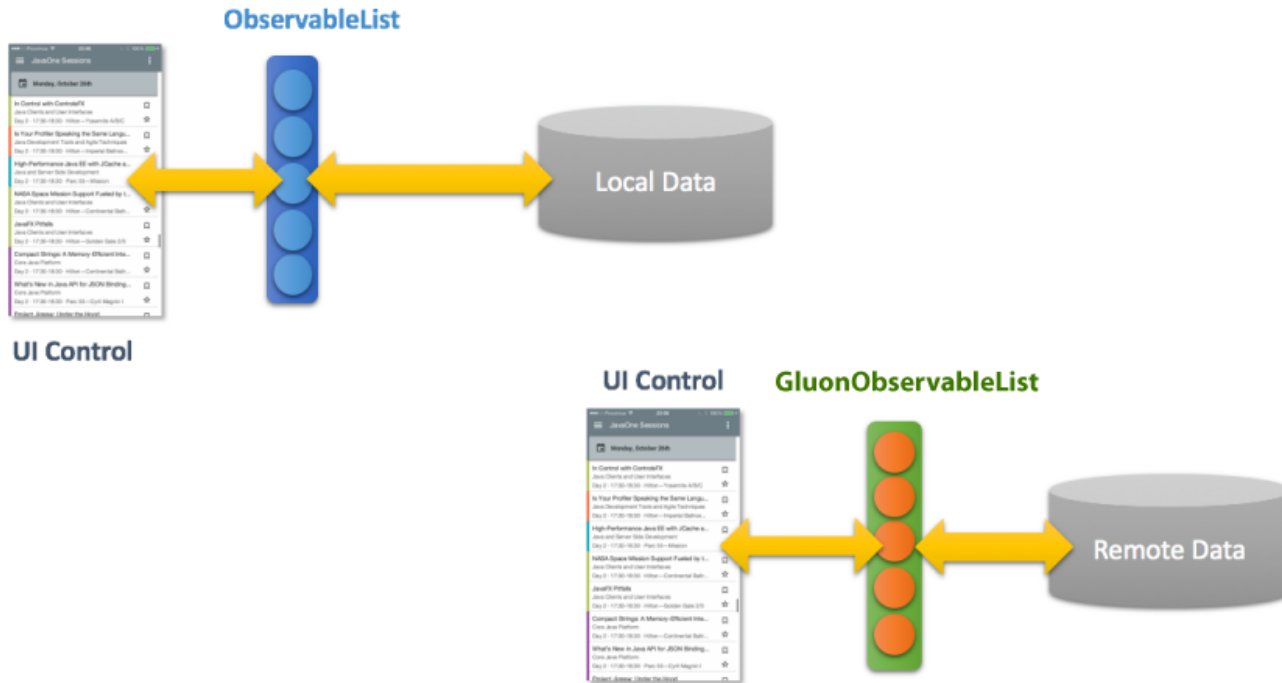
Afterburner Framework

- ▶ What is Afterburner?
 - Lightweight framework
 - Provides dependency injection
- ▶ Why Use Afterburner?
 - Injects FXML for views and Java objects
 - Generates Java boilerplate code
- ▶ Advantages
 - Reduces Java code
 - Safe and easy object sharing among views

Gluon Cloud Demo



Remote List Binding



Data Synchronization

- ▶ Write Through
 - Update remote copy when local data changes
 - **LIST_WRITE_THROUGH**
 - **OBJECT_WRITE_THROUGH**
- ▶ Read Through
 - Update local copy when remote data changes
 - **LIST_READ_THROUGH**
 - **OBJECT_READ_THROUGH**

Authentication Mode

- ▶ Credentials
 - Register application on CloudLink
 - Application keys supplied by Gluon
- ▶ Gluon Client

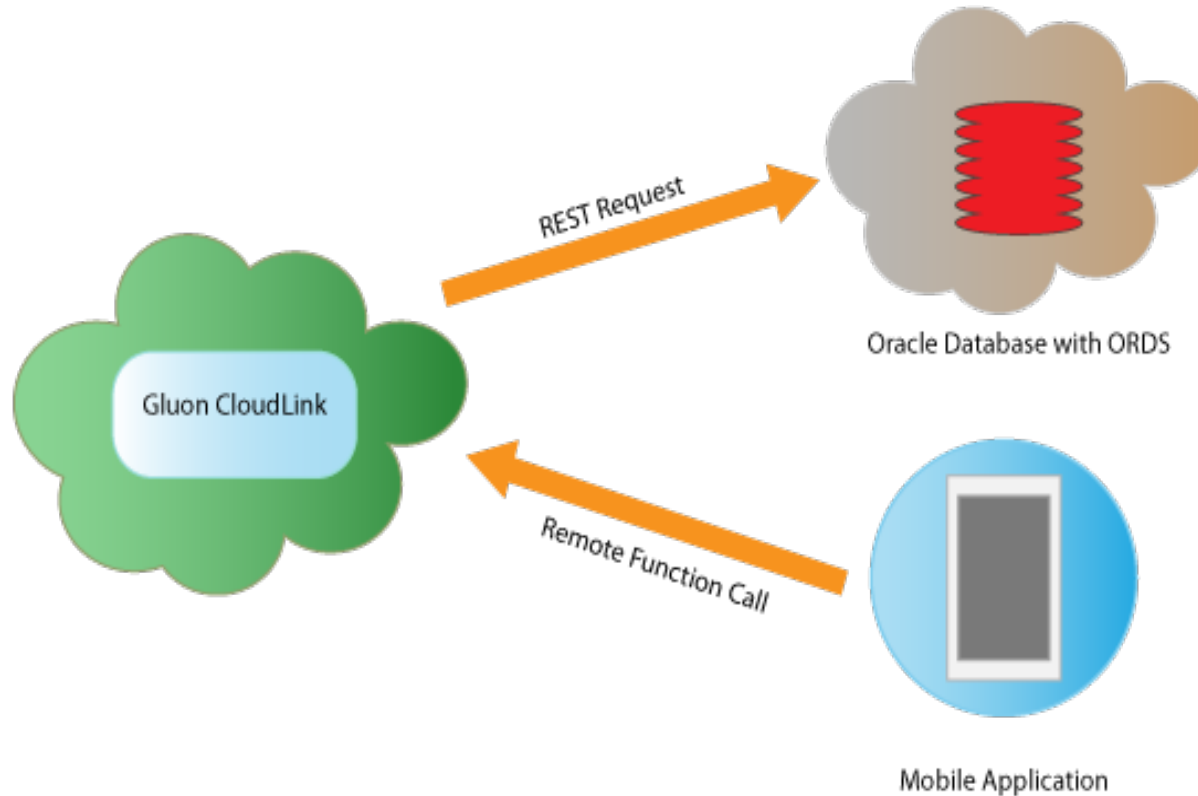
```
GluonClient gluonClient =  
    GluonClientBuilder.create().credentials(  
        new GluonCredentials(APPKEY, APPSECRET))  
        .authenticationMode(  
            AuthenticationMode.USER)  
        .build();
```


Data Provider

▶ Service Class

```
void getData() {  
    GluonObservableList<BPData> gluonBPData =  
        DataProvider.retrieveList(  
            gluonClient.createListDataReader(  
                user.get().getNick() +  
                user.get().getNetworkId(), BPData.class,  
                SyncFlag.LIST_READ_THROUGH,  
                SyncFlag.LIST_WRITE_THROUGH,  
                SyncFlag.OBJECT_READ_THROUGH,  
                SyncFlag.OBJECT_WRITE_THROUGH));  
    . . .  
}
```

Oracle Database



Oracle REST Data Services



HTTP(S) client



Oracle REST Data Services



Oracle Database

People Demo

Employee Display	
Dept: 10	
CLARK MANAGER Hired: 1981-06-09 Salary: 3000	
Harry Wizard Hired: 2018-10-17 Salary: 4200	
Hermione Wizard Hired: 2018-10-12 Salary: 4500	
KING PRESIDENT Hired: 1981-11-17 Salary: 5000	
MILLER CLERK Hired: 1982-01-23 Salary: 1300	
Dept: 20	
FORD ANALYST Hired: 1981-12-03 Salary: 3000	
JONES MANAGER Hired: 1981-04-02 Salary: 2975	
SCOTT ANALYST Hired: 1987-04-19 Salary: 3000	

Remote Functions with REST

REST Method	Remote Function
READ	getEmployees()
READ	getEmployee()
POST	createEmployee()
PUT	updateEmployee()
DELETE	deleteEmployee()

API Dashboard

API Management 👤

[Functions](#) [Authentication](#) [Call Log](#)

Remote Functions getEmployee + ✎ 🗑️

Function Information

Type: HTTP Request
Name: getEmployee
Internal only: false
Method: GET
End Point: http://54.217.216.239/ords/pdb1/myuser1/testmodule9/emp/\$empno
Request Body: none

Function Parameters ✕ Test + 🗑️

	Type	Name	Value	Test Value
<input type="checkbox"/>	Variable param	empno	7521	7369

Remote Functions

▶ GET Method Definition

```
public class RemoteService {
    public <T> GluonObservableObject<T>
        getEmployees(Class<T> clazz) {
        RemoteFunctionObject function =
            RemoteFunctionBuilder
                .create("getEmployees")
                .object();
        return function.call(clazz);
    }
    // other remote functions. . .
}
```

Remote Functions

▶ Dependency Injection

```
@Inject
private RemoteService remoteService;

@FXML
private
CharmListView<Employee, Integer> charmListView;

private
GluonObservableObject<WorkForce> workforce;

private ChangeListener<? super Boolean>
        listener = null;
```


Remote Functions

▶ GET Method Invocation

```
workforce = remoteService.getEmployees(WorkForce.class);  
.  
.  
workforce.initializedProperty().addListener(  
    listener = (ObservableValue<? extends Boolean>  
        obsrv, Boolean ov, Boolean nv) -> {  
        if (nv) {  
            charmListView.setItems(FXCollections.  
                observableArrayList(  
                    workforce.get().getItems()));  
        }  
    });
```

Remote Functions

▶ DELETE Method Definition

```
public class RemoteService {
    public <T> GluonObservableObject<T>
        deleteEmployee(Class<T> clazz, String value) {
            RemoteFunctionObject function =
                RemoteFunctionBuilder
                    .create("deleteEmployee")
                    .param("empno", value)
                    .object();
            return function.call(clazz);
        }
}
```

Summary

▶ JavaFX Advantages

- Platform independent source code
- Observables, binding & background tasks to sync UI
- Flexible skinning to fit mobile form

▶ Useful Frameworks

- Gluon/Gradle framework for mobile deployment
- Based on OpenJDK
- Afterburner framework for dependency injection
- Oracle Cloud Database and ORDS

Wrap Up

▶ Thanks for Coming!

- paul@asgteach.com
- gail@asgteach.com

@paul_asgteach
@gail_asgteach

▶ Source Code

- asgteach.com
 - SDJUG 2019 Meetup Examples
 - [Click to Download](#)
- Q & A

