



# Adobe Flex & Grails

## RIA, REST und XML



**Pascal Schudel**

Consultant

[pascal.schudel@trivadis.com](mailto:pascal.schudel@trivadis.com)

**Mischa Kölliker**

Principal Consultant

[mischa.koelliker@trivadis.com](mailto:mischa.koelliker@trivadis.com)

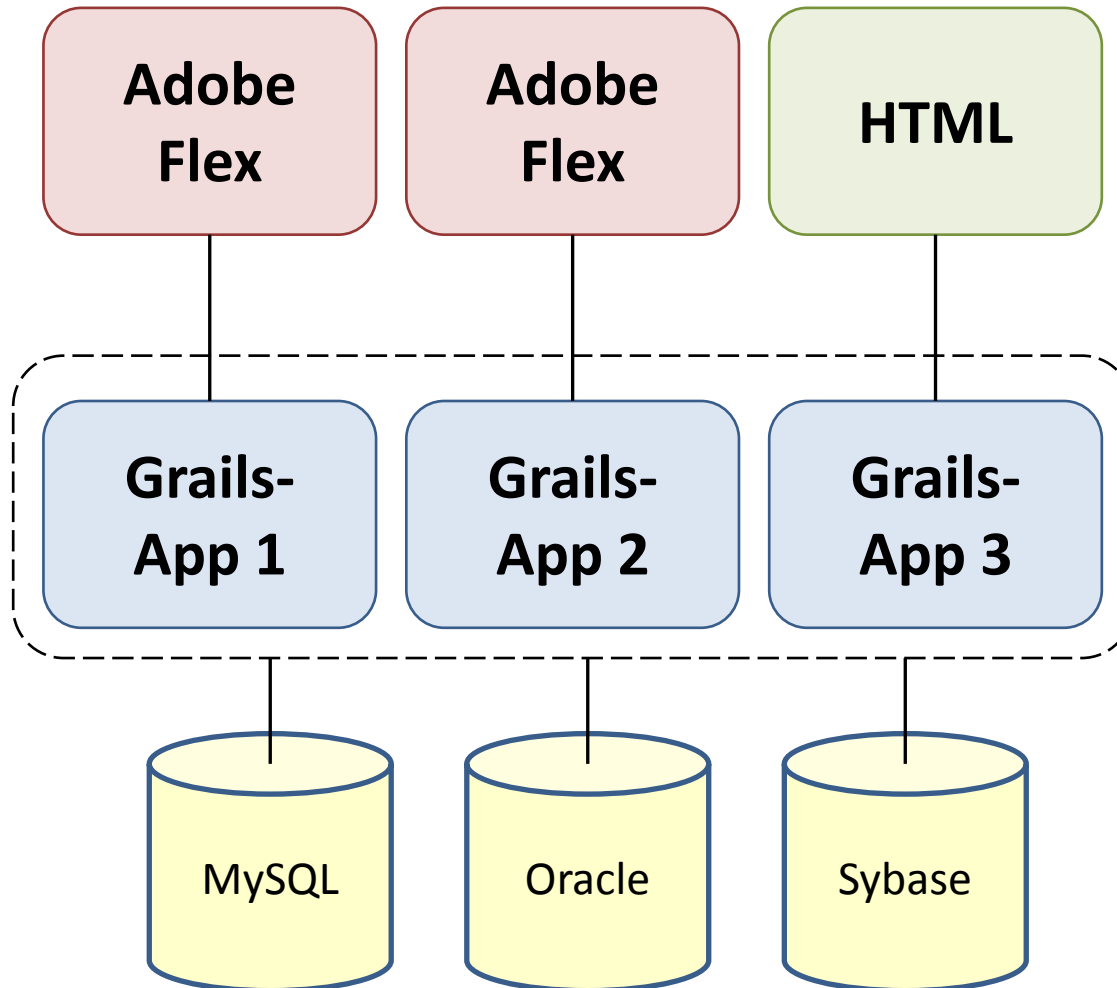
JUGS Zürich, 8. Juli 2010

**trivadis**

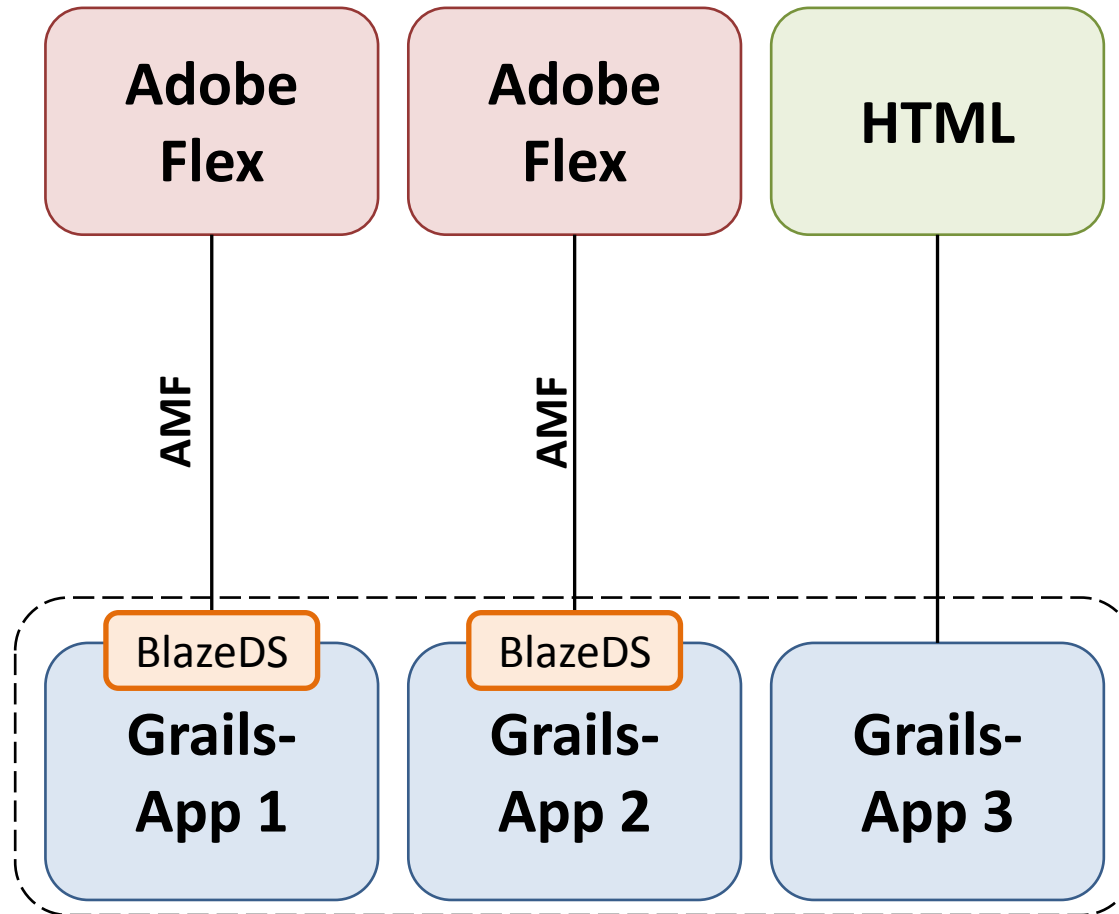
makes IT easier.



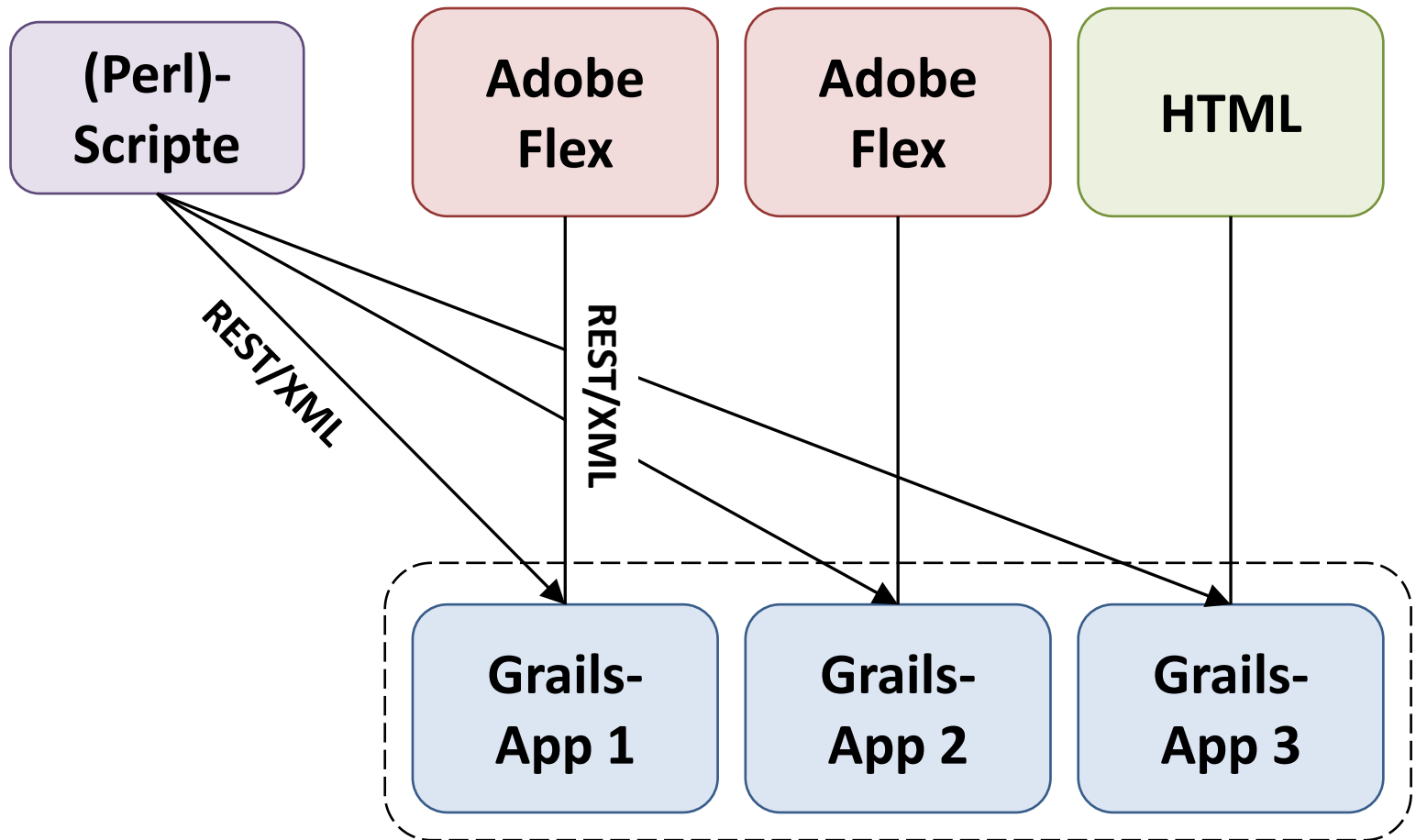
# Was haben wir gebaut?



# Der übliche Weg: BlazeDS und AMF



# Script-Clients: REST & XML





# Sehr heterogene Systemlandschaft

- Client

- Browser



- Betriebssysteme



- Server

- Datenbanken



# Flex & Grails – WTF?

## Roundtrip by example

## Holz- oder Königsweg?

# Wer wir sind

- Pascal Schudel

- Java Consultant
- Grails-Experte



- Mischa Kölliker

- Java & SOA Consultant
- Buchautor
- Architektur, Konzeption



- Trivadis AG

- Enterprise-Lösungen mit Java-, Oracle- und Microsoft-Technologien
- 550 Mitarbeiter an 11 Standorten in DACH





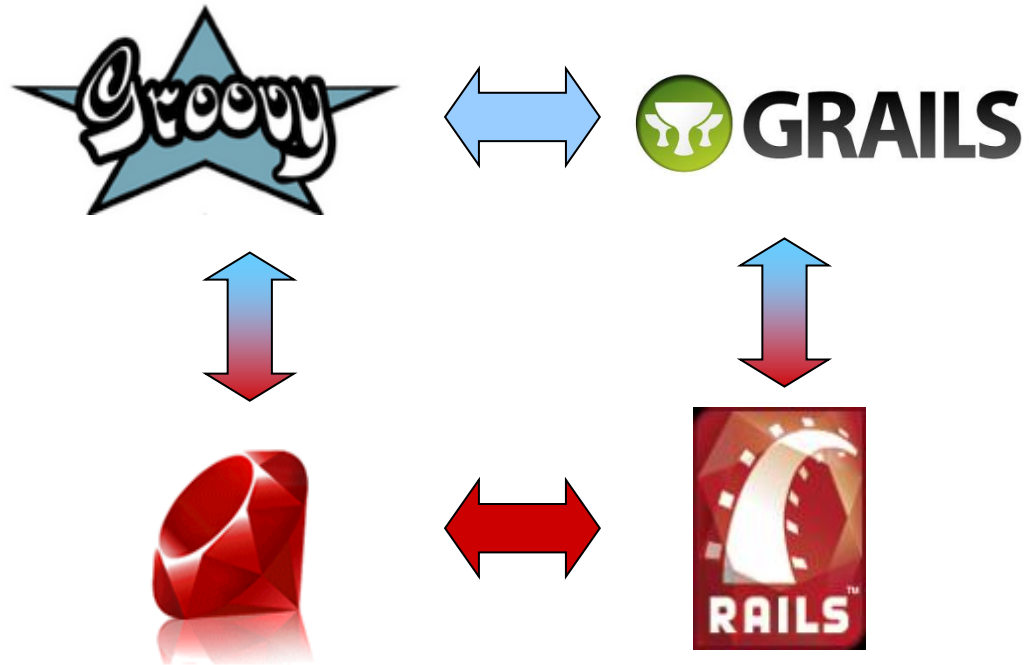
# ➤ Flex & Grails – WTF?

Roundtrip by example

Holz- oder Königsweg?

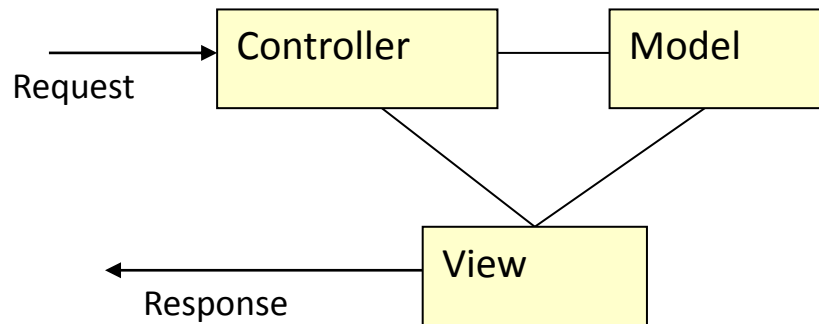
# Grails

# Grails vs. Rails

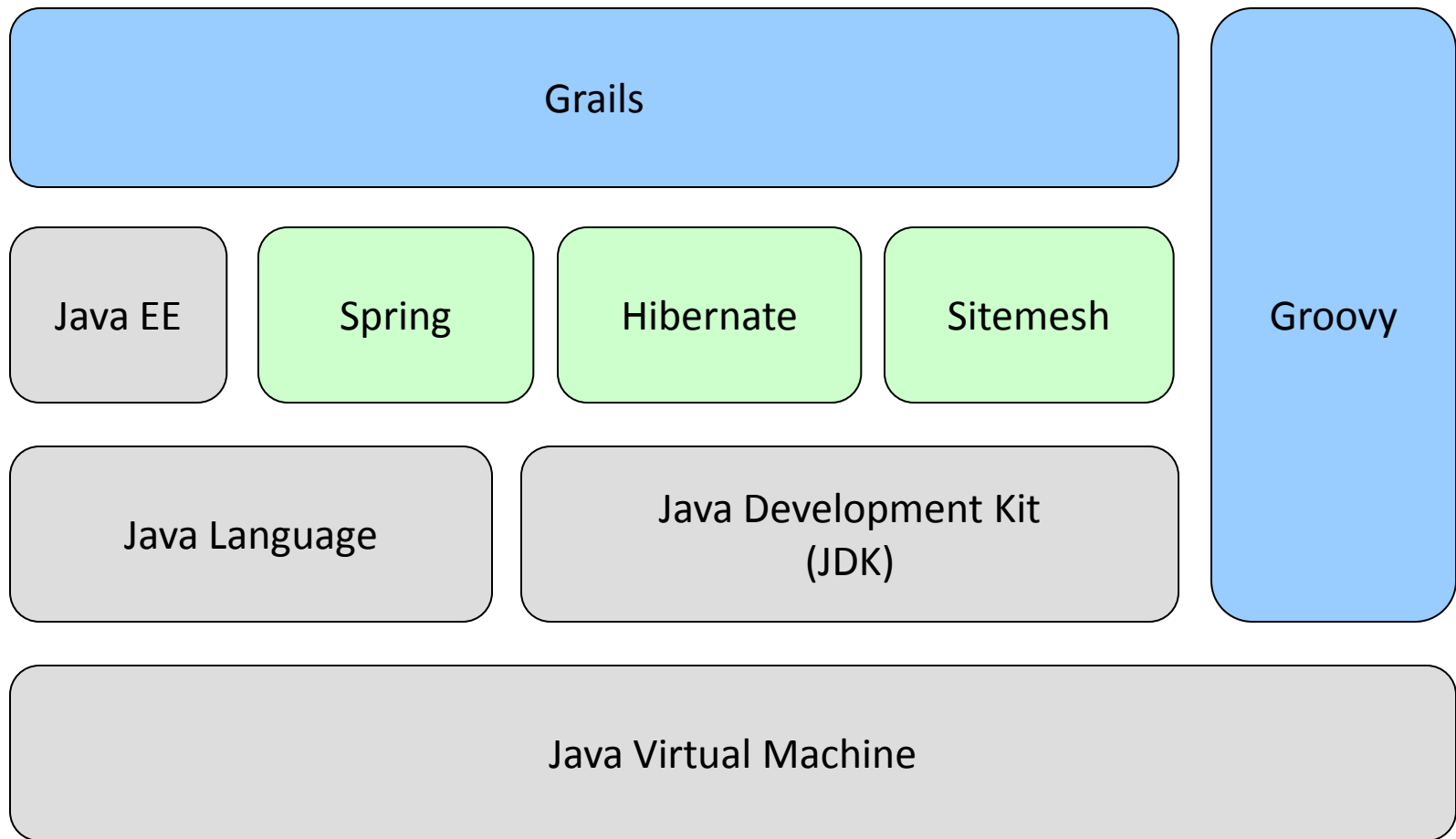


# Grails

- Open-source
- RIAD – Rapid Internet Application Development
- Action-basiert (im Gegensatz zu Komponenten-basiert)
- MVC-basiert

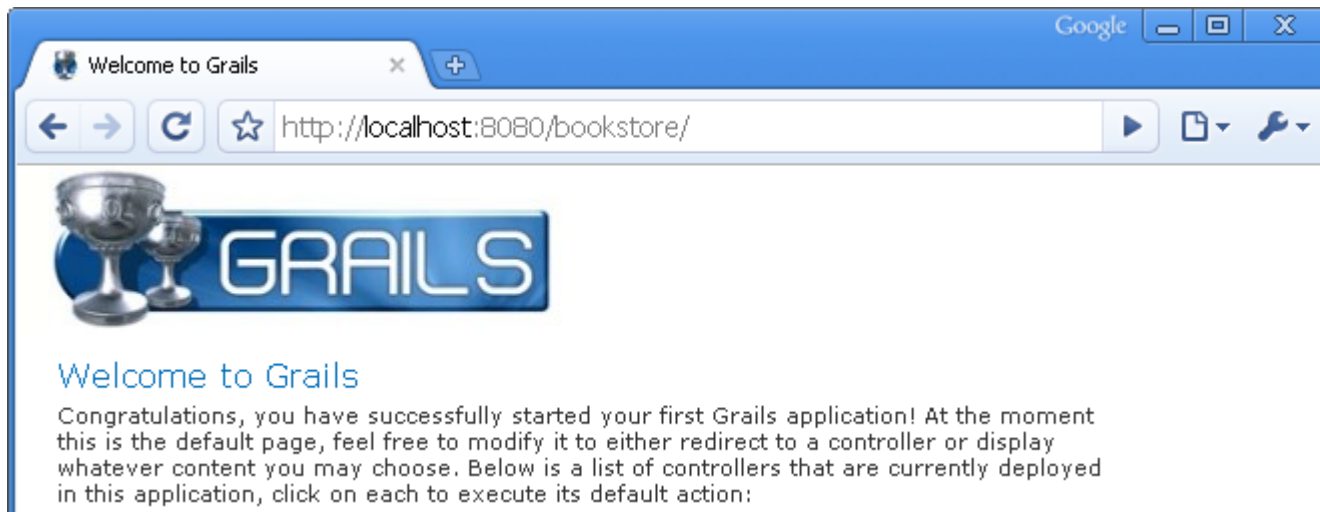
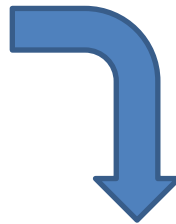
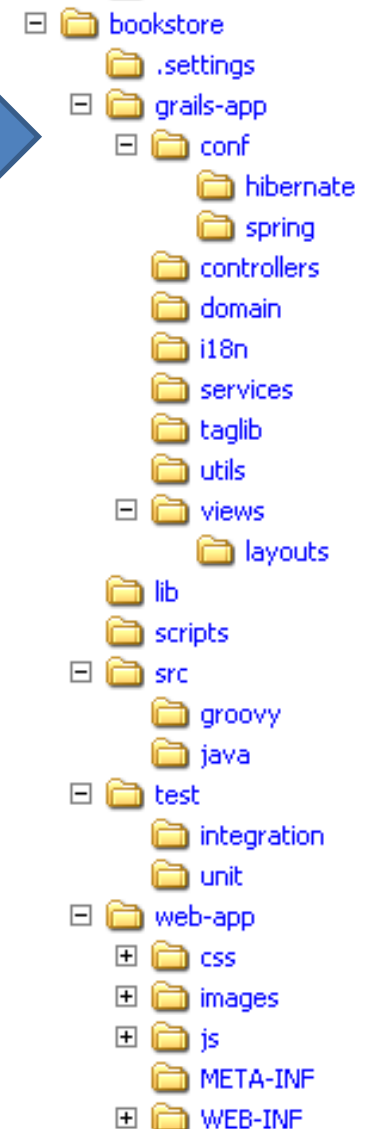


# Grails Technologiестack

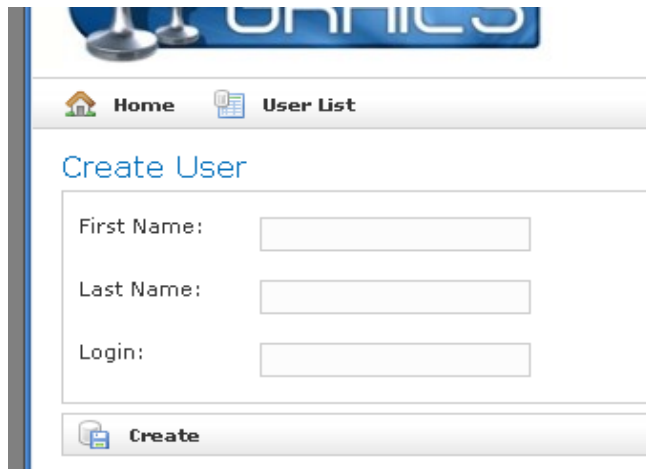
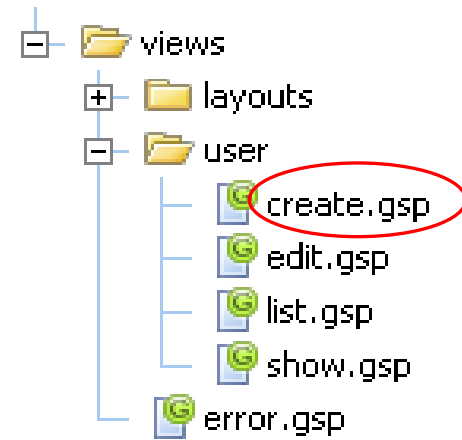
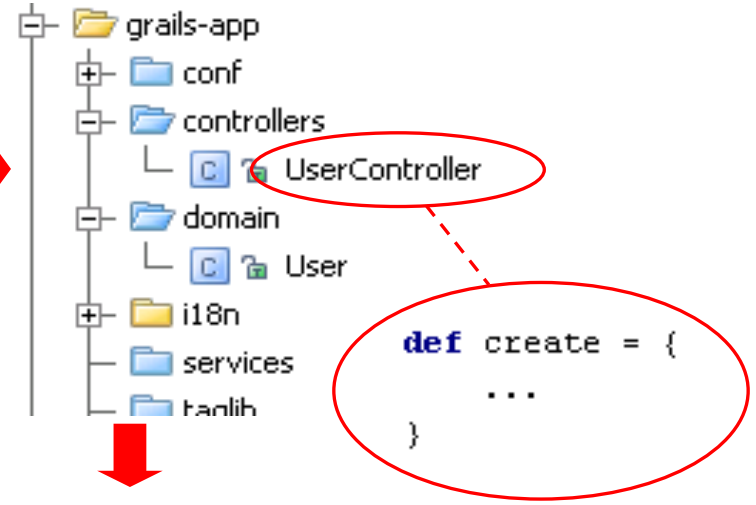


# Eine Grails-Anwendung erstellen

- > `grails create-app bookstore`
- > `cd bookstore`
- > `grails run-app`



# Codieren nach Konventionen



# Adobe Flex



# Die aktuellen Top 3 RIA-Frameworks

- Adobe Flex



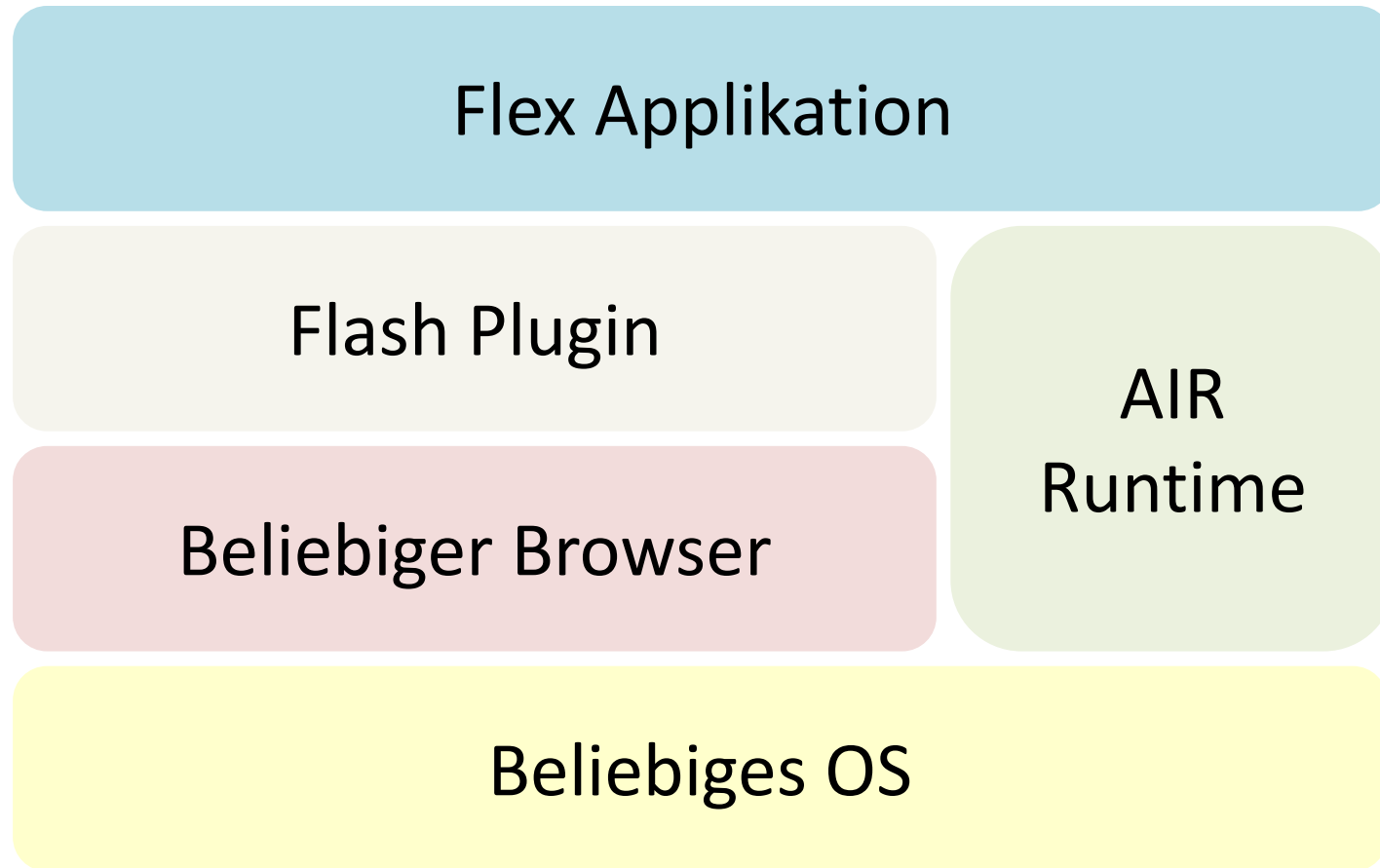
- Microsoft Silverlight



- JavaFX



# Adobe Flex Stack



# Ansprechendes Default-L&F

**Einfache Suche**

MAC Adressen meiner ISGs  
 Meine erfassten MAC Adressen

MAC

**Erweiterte Suche**

**Erfassen**    Gruppierung **Aktives Profil** Anzahl Einträge: 3102

Filterkriterium

MAC	ISG	Beschreibung	Host ID	Besitzer	Erstellt	Aktives Profil	Profile
▶ ee-iis-client							
▶ ee-iis-server							
▶ ee-iis-hb9zz							
▶ zo-spec							
▶ vseth-client							
▶ vseth-special							
▼ mavt3-client-1							
00-0e-7f-6c-c5-e2	mavt-ISG	ZFM	alberich	bcadonau	2008-12-04	mavt3-client-1	mavt3-client-1
00-0f-fe-79-25-67	mavt-ISG		30ffe792567	aholinger	2008-12-04	mavt3-client-1	mavt3-client-1
00-17-a4-99-57-de	mavt-ISG		0017a49957de	aholinger	2008-12-04	mavt3-client-1	mavt3-client-1
▶ sos-client							
▼ net-server							
00-03-93-19-57-5e	adm-net		00-03-93-19-57-5e	beatmue	2008-12-04	net-server	net-server
00-05-02-71-11-da	adm-net		00-05-02-71-11-da	derkv	2008-12-04	net-server	net-server
00-08-02-50-91-fc	adm-net		00-08-02-50-91-fc	zillert	2008-12-04	net-server	net-server
00-08-02-5b-f9-1c	adm-net		00-08-02-5b-f9-1c	zillert	2008-12-04	net-server	net-server
00-08-02-5d-f0-12	adm-net		00-08-02-5d-f0-12	beatmue	2008-12-04	net-server	net-server
00-08-c7-eb-b8-d4	adm-net		00-08-c7-eb-b8-d4	derkv	2008-12-04	net-server	net-server
00-0d-61-b8-36-d8	adm-net		00-0d-61-b8-36-d8	zillert	2008-12-04	net-server	net-server
00-13-72-33-76-02	adm-net		00-13-72-33-76-02	zillert	2008-12-04	net-server	net-server
00-15-c5-e2-61-02	adm-net		00-15-c5-e2-61-02	zillert	2008-12-04	net-server	net-server
00-30-65-c9-dc-40	adm-net		00-30-65-c9-dc-40	zillert	2008-12-04	net-server	net-server
00-30-65-ee-f4-d0	adm-net		00-30-65-ee-f4-d0	derkv	2008-12-04	net-server	net-server
00-50-8b-cd-ce-58	adm-net		00-50-8b-cd-ce-58	zillert	2008-12-04	net-server	net-server
00-c0-4f-61-1a-a4	adm-net		00-c0-4f-61-1a-a4	beatmue	2008-12-04	net-server	net-server

**Erfassen**

**Editieren**

**MAC Adresse editieren**

ISG \* **mavt-ISG**

**ISG nach VPZ auswählen**

Profile **Zugewiesene Profile** **Mögliche Profile**

**mavt1-hg-client225**  **baug-igp**

VPN Zugriff Ja  VPN Zugriff Ja

**baug-igt**

VPN Zugriff Ja

**baug-ivt**

VPN Zugriff Ja

**baug-karto**

Aktives Profil \* **mavt1-hg-client225**

Beschreibung

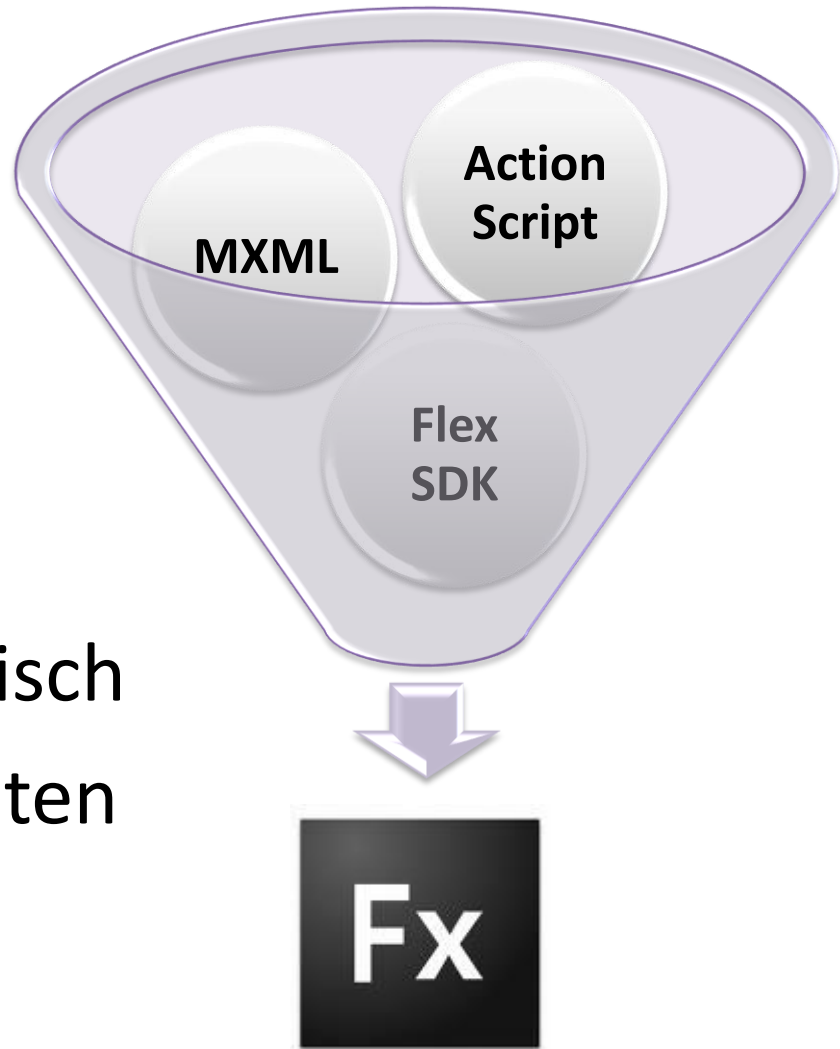
Host ID

MAC Adresse \*

**Bitte MAC Adresse im Format xx-xx-xx-xx-xx-xx eingeben (Beispiel: 00-1C-23-74-5F-C4)**

**Dateitransfer**

- MXML: statisch
- ActionScript: dynamisch
- Flex SDK: Komponenten



```

<mx:Canvas label="Einfache Suche" width="100%" height="100%" id="predefinedSearchPanel"
    horizontalScrollPolicy="off" verticalScrollPolicy="off">

    <mx:RadioButtonGroup id="searchRadioGroup"/>
    <mx:RadioButton x="10" y="10" label="Meine erfassten MAC Adressen"
        groupName="searchRadioGroup"
        selected="true" id="searchPredefinedUserMacs" click="searchPredefined()"/>
    <mx:RadioButton label="MAC Adressen meiner ISGs" groupName="searchRadioGroup"
        id="searchPredefinedIsgMacs" x="10" y="40" click="searchPredefined()"/>
    <mx:Form x="10" y="70" id="form2"
        defaultButton="{searchPredefinedSubmitButton}" paddingLeft="5">
        <mx:FormItem tooltip="Geben Sie die gesuchte MAC Adresse ein (Wildcard %)" direction="vertical">
            <mx:Label text="MAC" id="searchPredefinedMacField"/>
            <mx:TextInput width="170" id="queryPredefinedMacText"/>
        </mx:FormItem>
        <mx:FormItem>
            <mx:HBox>
                <mx:Button label="Suche" id="searchPredefinedSubmitButton"
                    click="searchPredefined();"
                    enabled="true"/>
            </mx:HBox>
        </mx:FormItem>
    </mx:Form>
</mx:Canvas>

```

```

847 private function getAssignableCurrentProfiles():XMLListCollection {
848     var profiles:XMLListCollection = getMacEditItemProfiles();
849     for each (var userProfile:XML in profileUserList) {
850         var isListed:Boolean = false;
851         for each (var assignableProfile:XML in profiles) {
852             if (assignableProfile.id == userProfile.id) {
853                 isListed = true;
854                 break;
855             }
856         }
857         if (!isListed) {
858             profiles.addItem(userProfile.copy());
859         }
860     }
861     return profiles;
862 }

```



```

<mx:Canvas label="Einfache Suche" width="100%" height="100%" id="predefinedSearchPanel"
    horizontalScrollPolicy="off" verticalScrollPolicy="off">

<mx:RadioButtonGroup id="searchRadioGroup"/>
<mx:RadioButton x="10" y="10" label="Meine erfassten MAC Adressen"
    groupName="searchRadioGroup"
    selected="true" id="searchPredefinedUserMacs" click="searchPredefined(
<mx:RadioButton label="MAC Adressen meiner ISGs" groupName="searchRadioGroup"
    id="searchPredefinedIsgMacs" x="10" y="40" click="searchPredefined()"/>
<mx:Form x="10" y="70" id="form2"
    defaultButton="{searchPredefinedSubmitButton}" paddingLeft="5">
  <mx:FormItem tooltip="Geben Sie die gesuchte MAC Adresse ein (Wildcard %)" direction="vertical">
    <mx:Label text="MAC" id="searchPredefinedMacField"/>
    <mx:TextInput width="170" id="queryPredefinedMacText"/>
  </mx:FormItem>
  <mx:FormItem>
    <mx:HBox>
      <mx:Button label="Suche" id="searchPredefinedSubmitButton"
        click="searchPredefined();"
        enabled="true"/>

```

```

852         if (assignableProfile.id == userProfile.id) {
853             isListed = true;
854             break;
855         }
856     }
857     if (!isListed) {
858         profiles.addItem(userProfile.copy());
859     }
860 }
861 return profiles;
862 }

```

```

<mx:Canvas label="Einfache Suche" width="100%" height="100%" id="predefinedSearchPanel"
    horizontalScrollPolicy="off" verticalScrollPolicy="off">

<mx:RadioButtonGroup id="searchRadioGroup"/>
<mx:RadioButton x="10" y="10" label="Meine erfassten MAC Adressen"
    groupName="searchRadioGroup"
    selected="true" id="searchPredefinedUserMacs" click="searchPredefined()"/>
<mx:RadioButton label="MAC Adressen meiner ISGs" groupName="searchRadioGroup"
    id="searchPredefinedIsgMacs" x="10" y="40" click="searchPredefined()"/>
<mx:Form x="10" y="70" id="form2"
    defaultButton="{searchPredefinedSubmitButton}" paddingLeft="5">

```

```

private function getAssignableCurrentProfiles():XMLListCollection {
    var profiles:XMLListCollection = getMacEditItemProfiles();
    for each (var userProfile:XML in profileUserList) {
        var isListed:Boolean = false;
        for each (var assignableProfile:XML in profiles) {
            if (assignableProfile.id == userProfile.id) {
                isListed = true;
                break;
            }
        }
        if (!isListed) {
            profiles.addItem(userProfile.copy());
        }
    }
    return profiles;
}

```



Action  
Script

# Flex & Grails – WTF?

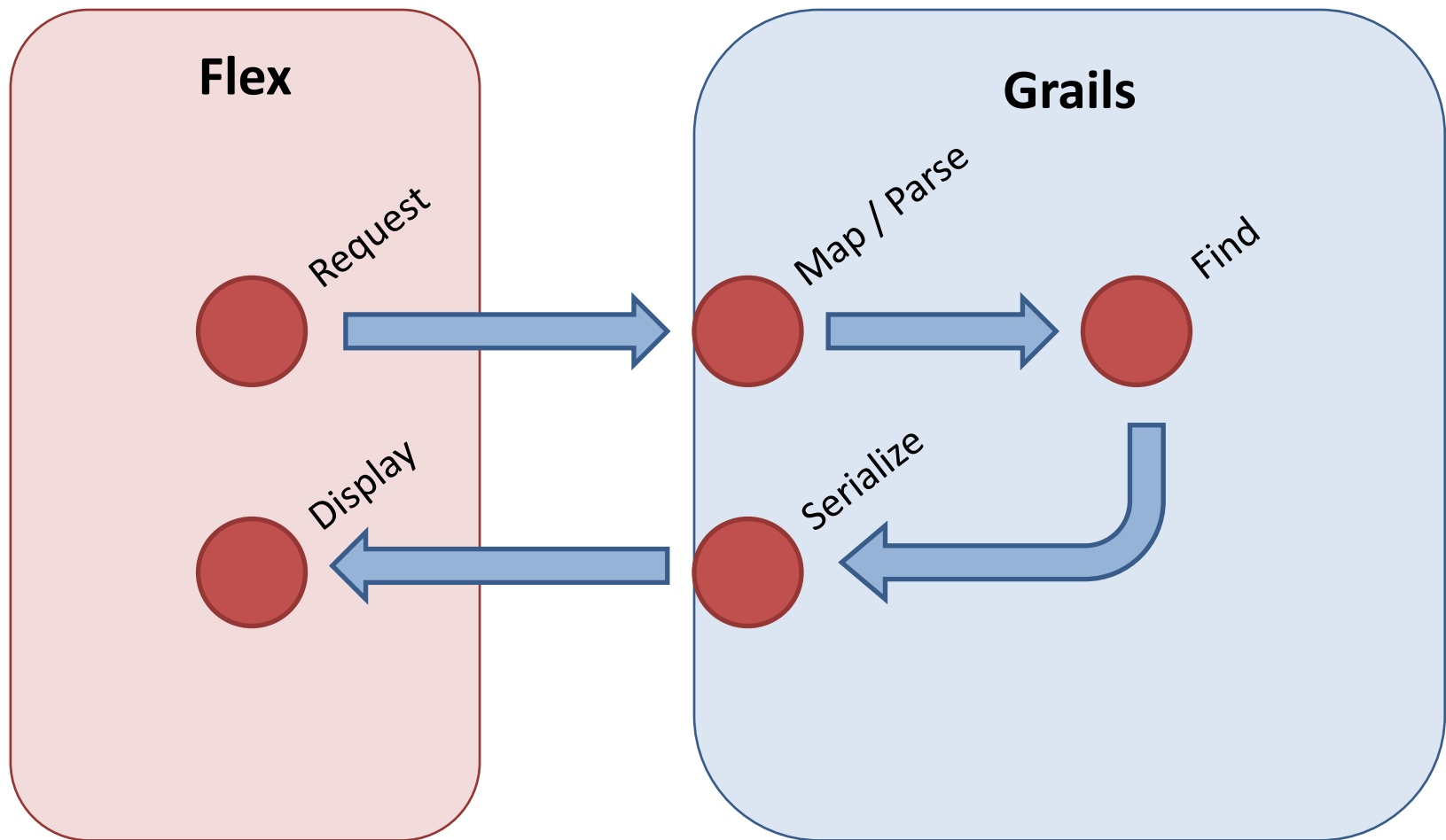


Roundtrip by example

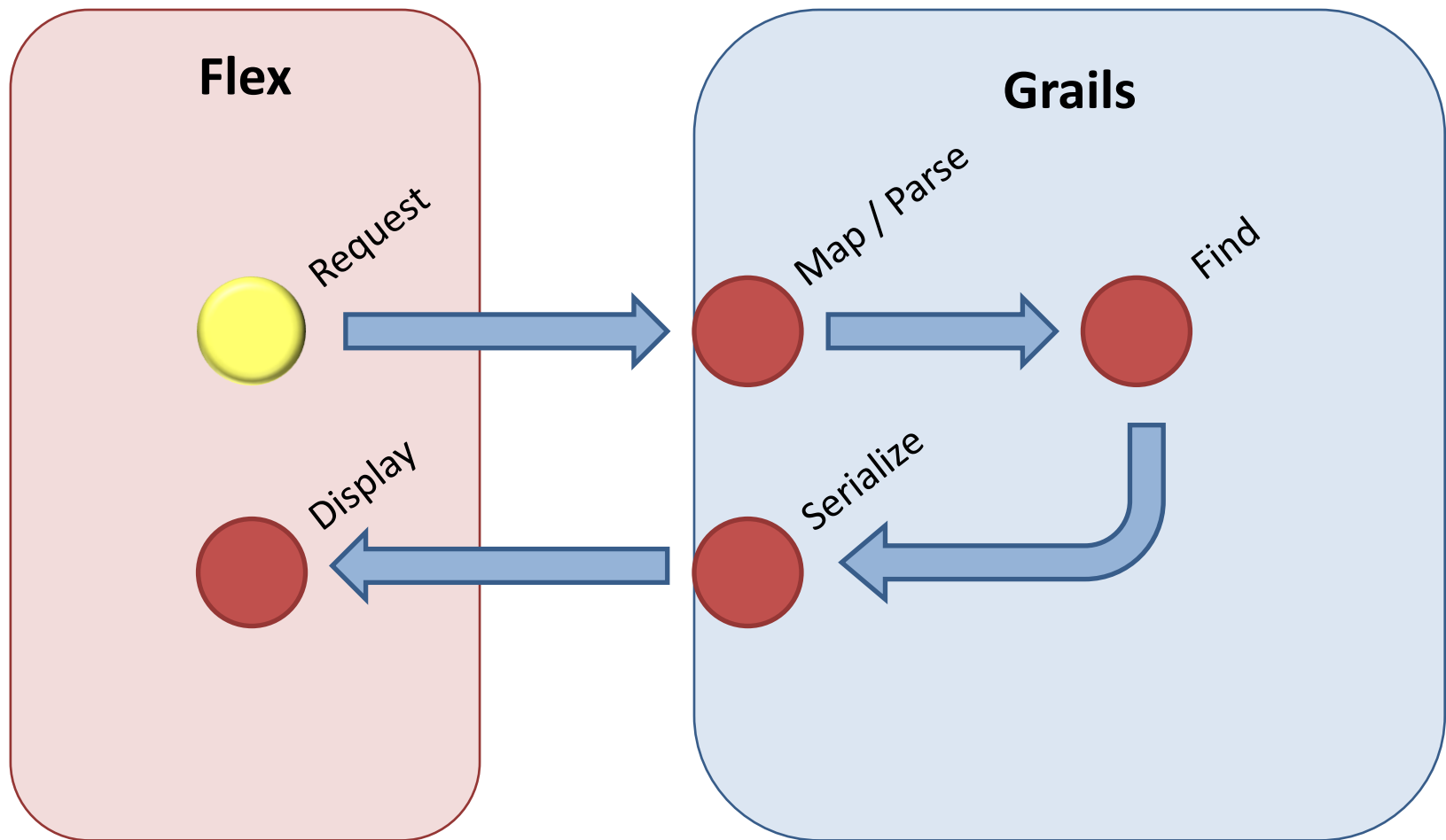
Holz- oder Königsweg?



# Roundtrip by example



# Roundtrip by example



# Kommunikation zum Server mit HTTPService

- Service-Stub-Definition in Services.mxml:

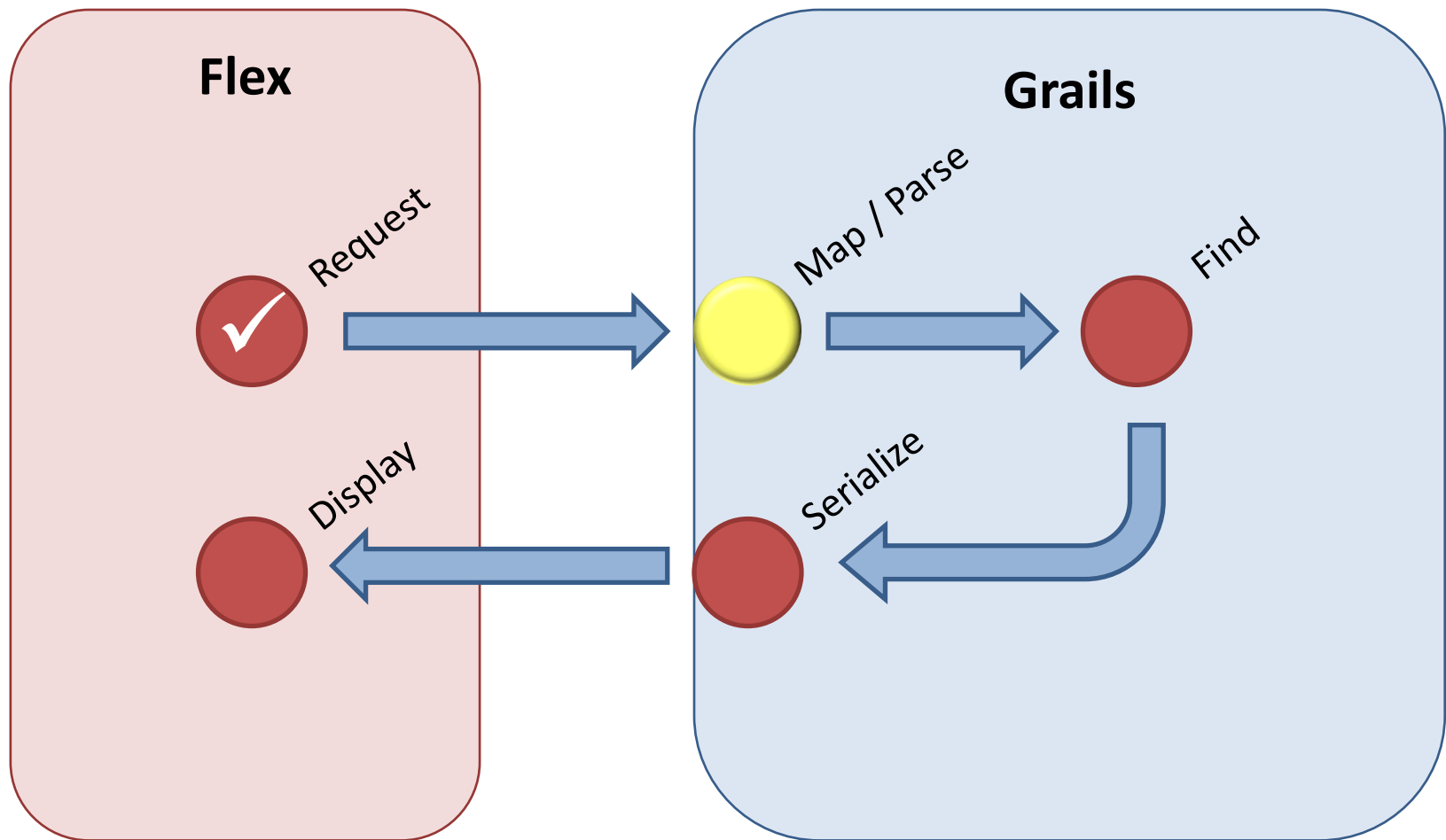
```
<mx:HTTPService id="loadNameToIPListService"
  contentType="application/xml" method="POST"
  url="http://server/ip/nameToIP/loadNameToIPList"
  resultFormat="e4x" />
```

- Service aufrufen:

```
var service:HTTPService = ServiceLocator.getHTTPService(name);
var parameters:XML;
parameters = <query>
    <nameToIP>{searchTerm}</nameToIP>
    <ipOrAlias>{ipOrAlias}</ipOrAlias>
    <example>{queryByExample}</example>
</query>;

service.send(parameters).addResponder(this);
```

# Roundtrip by example



# Grails Url-Mapping

- REST URL Mapping
  - URL Mapping mit Groovy DSL definieren

```
class UrlMappings {
    static mappings = {
        "/*controller/*action?/*id?" (parseRequest:true) {
            constraints {
                // apply constraints here
            }
        }
        "500" (view: '/error')

        "/rest/mac/*id?" (controller:"mac", parseRequest:true) {
            action = [GET:"show", PUT:"update", DELETE:"delete", POST:"save"]
        }
    }
}
```

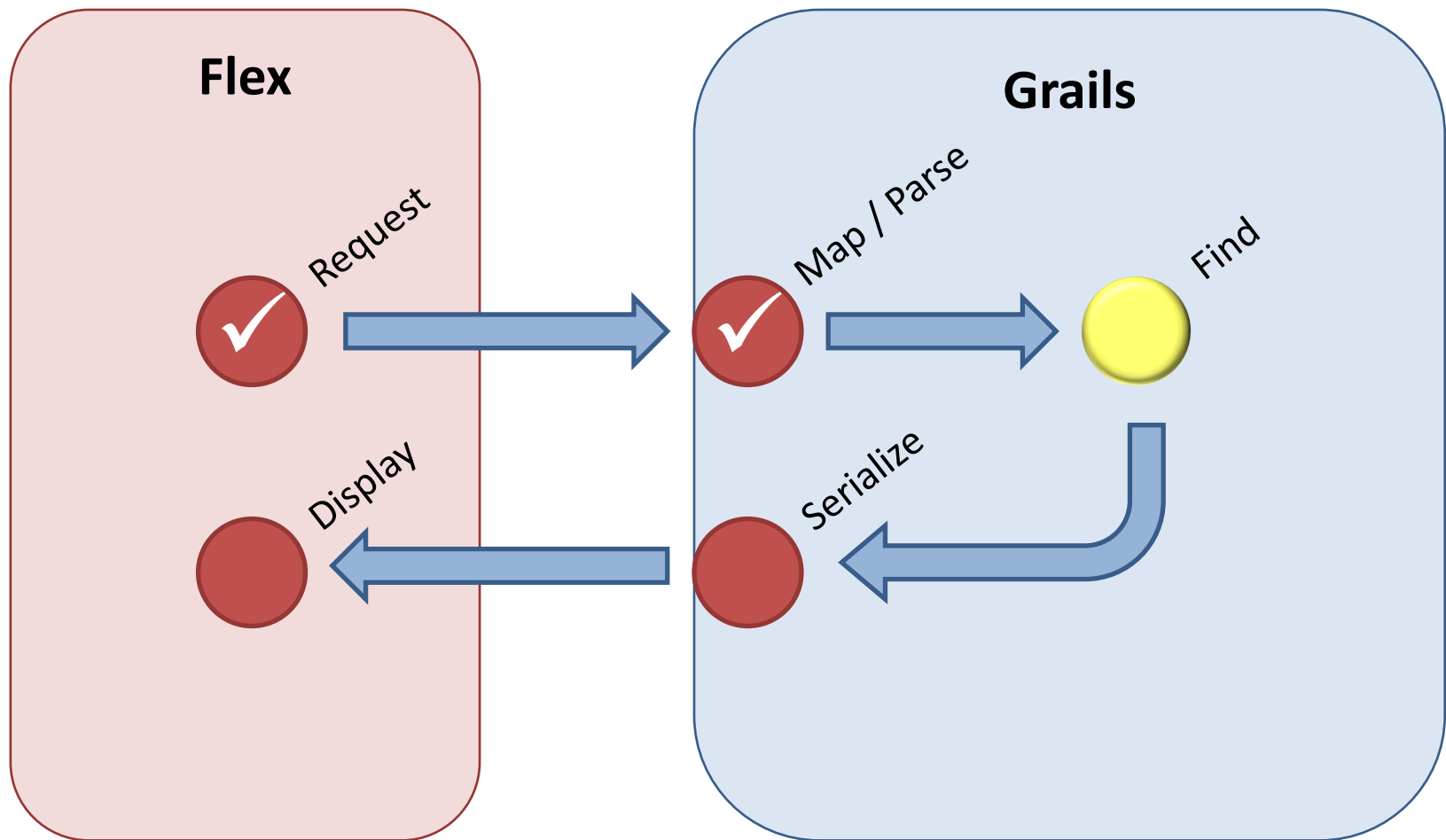
# XML parsen

- Convenience inkl. Databinding

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<people>
  <person>
    <name>Peter</name>
    <vorname>Meyer</vorname>
    <address id="12">
      ...
    </address>
  </person>
</people>
```

- XML parsen mit XmlSlurper (Groovy/Grails)
  - request.XML + GPath (ähnlich wie E4X)
    - def xml = request.XML
    - xml.people.each { p -> ... }
  - Flache XML-Struktur wird in älteren Grails Versionen automatisch in eine Map geparkt
    - Ab Grails 1.1 aktivierbar über URL Mapping

# Roundtrip by example



# Grails Finder

- Dynamische Finder-Methoden

- Performance verschlechtert sich bei vielen Aufrufen

```
Person person = Person.findByFirstName("Fritz")
```

```
Person person = Person.findByFirstNameAndLastName("Fritz", "Kueenzli")
```

- Alternative Finder

- Hibernate-Criteria API

```
def criteria = Person.createCriteria()
```

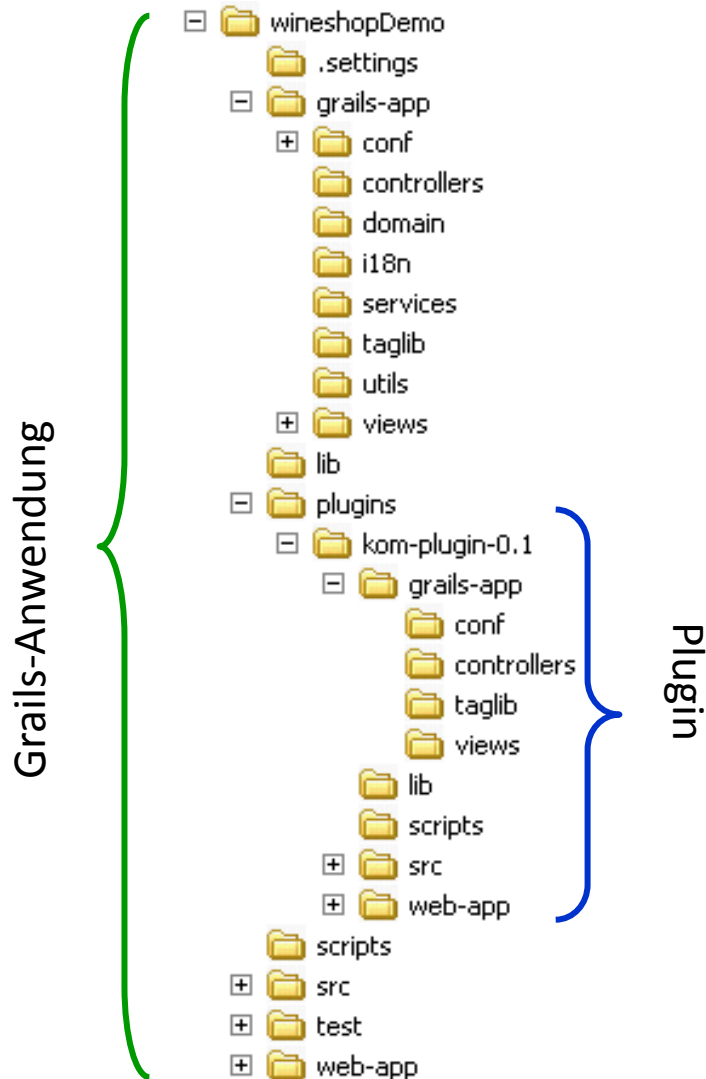
```
def personen = criteria {  
    like("firstName", "Fritz")  
    like("lastName", "Kueenzli")  
    eq("employeeNumber", 123456)  
}
```

- Hibernate-Queries

```
def personen = Person.findAll("from Person as p where p.firstName like 'Fritz'")
```



# Grails Plugin-System



- Abgespeckte Grails-Anwendungen
- Der Plugin-Inhalt wird in die Ordner der Grails-Anwendung kopiert (zur Laufzeit)
- Über 400 Plugins verfügbar

# Suchen mit Searchable-Plugin I/II

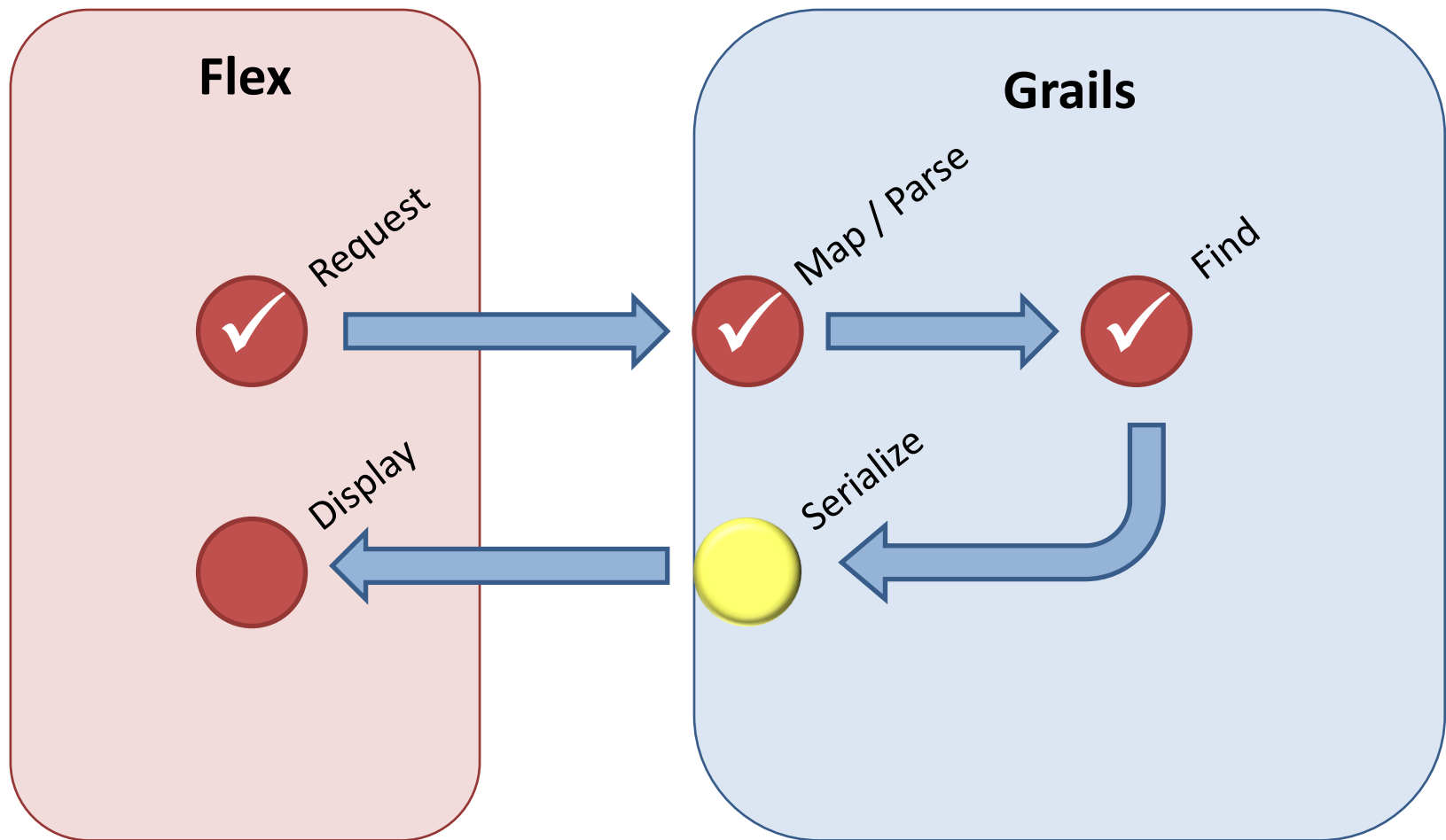
- Volltextsuche
- Basiert auf Compass und Apache Lucene
- Pro Domain-Klasse konfigurierbar welche Attribute im Index vorhanden sein sollen

```
rufnummer boost: 3.0  
apparatName boost: 2.0  
apparatTyp boost: 3.0  
location boost: 3.0
```

# Suchen mit Searchable-Plugin II/II

```
searchResult = ResourceView.search({
    if (queryMap?.status?.length() > 0) {
        term("status", queryMap.status)
    }
    if (queryMap.profil?.length() > 0) {
        must(wildcard("profilName", "${queryMap.profil}"))
    }
    if (queryMap.search?.length() > 0) {
        queryString(queryMap.search)
    }
}, ["max": MAX_RESULT, "reload": false])
```

# Roundtrip by example



# Daten an Client zurück senden

- XML serialisieren

- „Convenience“

`render person as XML`

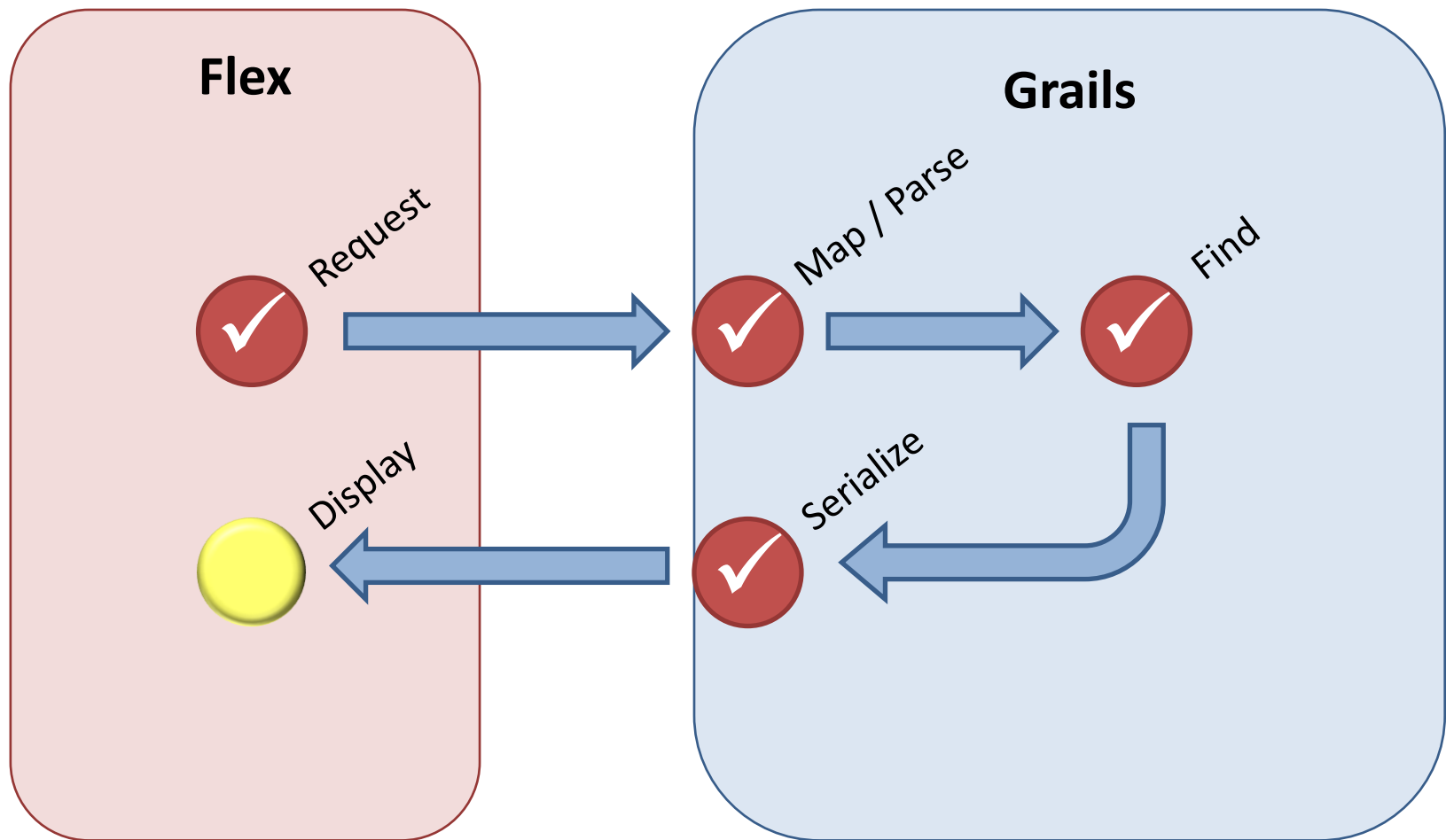
- ~~Builder~~

```
def show = {  
    Person p = Person.get(params.id)  
  
    render(contentType:"text/xml") {  
        person {  
            name(p.name)  
            vorname(p.vorname)  
        }  
    }  
}
```

- Template (GStringTemplate)

```
<?xml version="1.0"?>  
<people><% people.each { p -> %>  
    <person>  
        <id>${p.id ?: ""}</id>  
        <name>${p.name ?: ""}</name>  
        <vorname>${p.vorname ?: ""}</vorname>  
    </person><% } %>  
</people>
```

# Roundtrip by example



# Response verarbeiten mit HTTPService

Antwort direkt im DataGrid (Tabelle) anzeigen:

```
<mx:AdvancedDataGrid id="nameToIpResultList"
    dataProvider="{loadNameToIpListService.lastResult.nameToIp}">
    <mx:AdvancedDataGridColumn headerText="IP-Address"
        dataField="ip"/>
    <mx:AdvancedDataGridColumn headerText="Subnet"
        dataField="ipSubnet"/>
    <mx:AdvancedDataGridColumn headerText="FQ-Name"
        dataField="fqName"/>
    ...
</mx:AdvancedDataGrid>
```

# Anzeige des XML im DataGrid

IP	Prefix	Type	Name	DHCP	DHCP lease time [s]			Default Sub-/Domain	DDNS
					Min	Max	Default		
1:620:8:2000::	64	Subnet	id-kom.fw-in.1		3600	604800	604800	euro	
1:620:8:2000::	120	Subnet_Static	id-kom.fw-in.1-miau		3600	604800	604800	eurobiosyn.net	
								eurobiosyn.org	
								euroconstruct.ch	
								euromr.com	
								euromr.org	



# Response verarbeiten mit HTTPService

Sie erinnern sich: Service-Aufruf

```
service.send(parameters).addResponder(this);
```

Antwort verarbeiten:

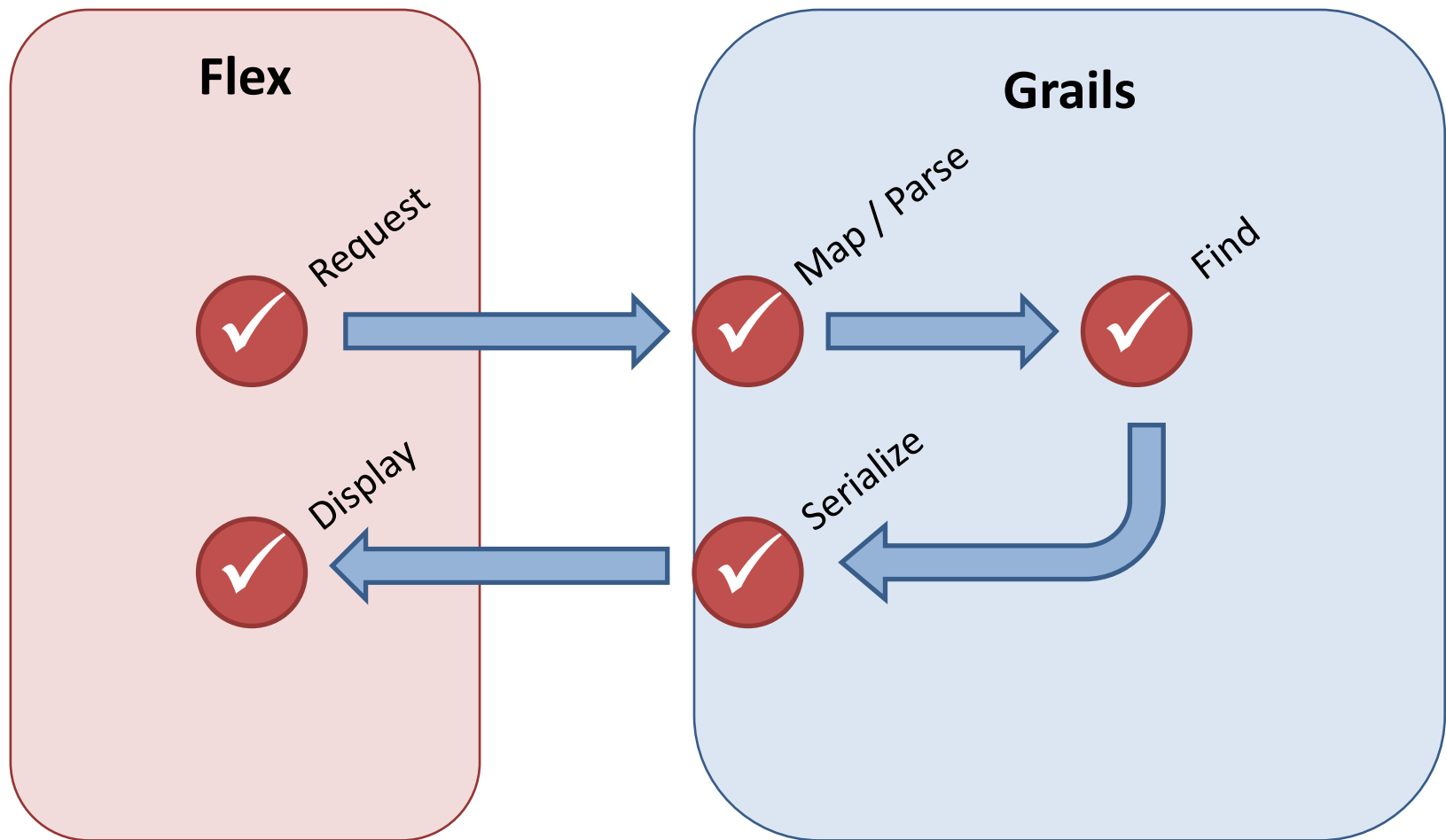
```
public override function result(data:Object):void
{
    var result:XML = XML(data.result);

    var resultList:XMLList = new XMLList(result.nameToIP);

    var origItemList:XMLList = resultList.source..net.(@id == mItem.@id);

    ...
}
```

# Roundtrip by example



# Flex & Grails – WTF?

Roundtrip by example

➤ Holz- oder Königsweg?

# Grails und Flex für Um-/Einsteiger geeignet

- Kunde hat grosses Wissen in Perl, PL/SQL und Oracle Forms
- Konnte nach kurzer Zeit im Projekt mitarbeiten

# Ein paar RESTriktionen

- Browser Design-Beschränkung:
  - Nur GET- und POST-Requests möglich
- Mögliche Workarounds:
  - Alternative URL: `/nameToIp/save/123`
  - BlazeDS als Proxy (`useProxy=„true“`)
  - Seit Grails 1.2
    - HTTP Header `X-HTTP-Method-Override`
    - URL-Parameter `_method`

# Entwicklungsumgebungen / Lizenzkosten

- Adobe Flash Builder (früher Flex Builder, Eclipse Plugin)
  - GUI-Designer
  - Code Completion
  - Debugger
  - Lizenzkosten: Fr. 0.- / Fr. 300.- / Fr. 890.-
- IntelliJ IDEA
  - Aktuell beste Unterstützung für Grails
    - Code Completion (bspw. auch für dynamische Finder)
    - Wizards für Controller, Services
  - Kein GUI-Builder für Flex
  - Lizenzkosten: \$99 / \$249 / \$599

# Fazit

- Flex
  - Sehr ausgereifte GUI-Komponenten
  - Open-Source
- Grails
  - Rapid Prototyping
  - Grosse und sehr aktive Community
  - Dank Plugins fast unendliche Erweiterbarkeit
  - Performance ok
- Flex und Grails ergänzen sich in der Praxis sehr gut