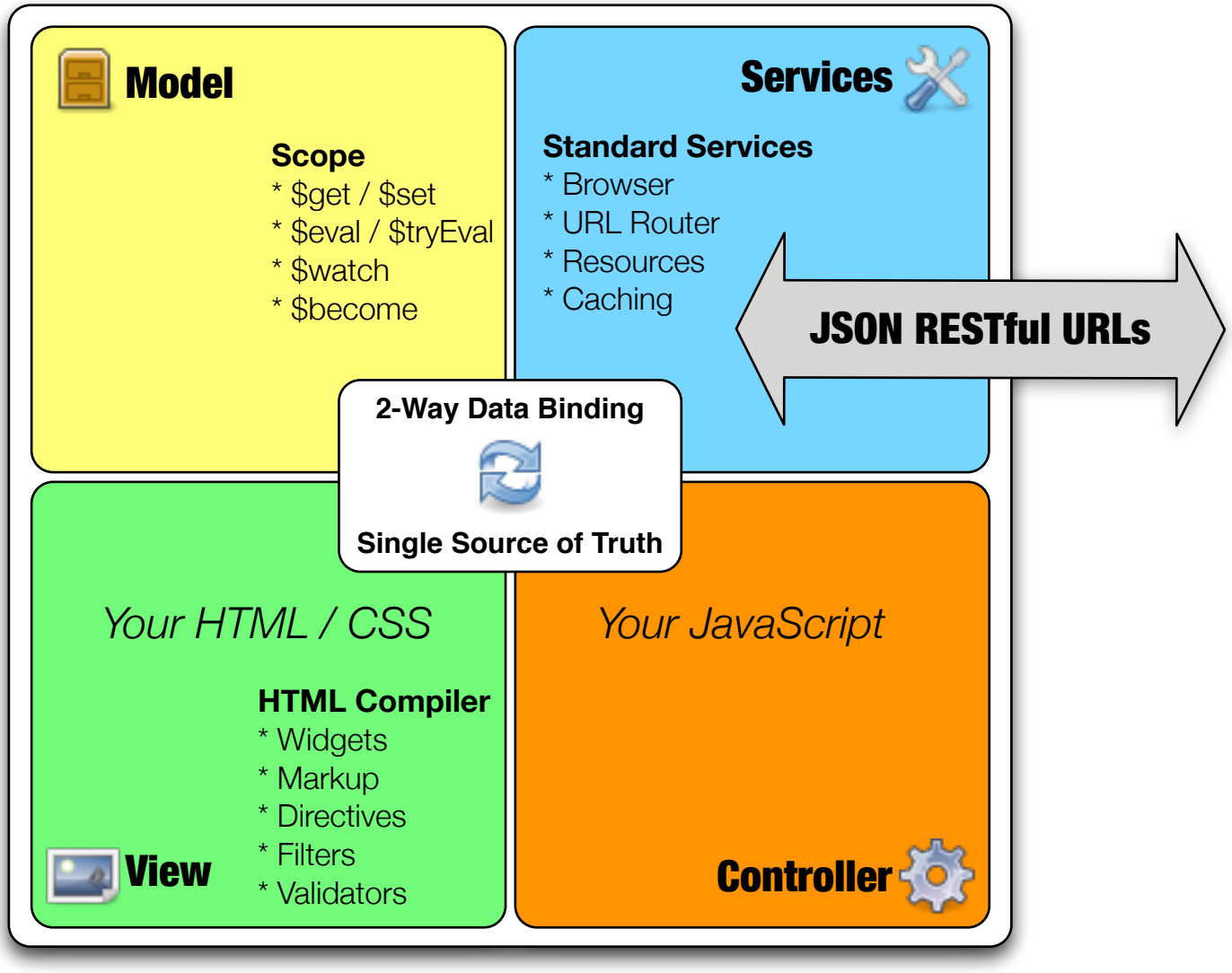
Guiding Principles

* Convention over Configuration
* Declarative / Self Describing
* Testable
* DRY (Don't Repeat Yourself)
* CRUD ~ 80% -> make it trivial

<angular/> Enabled Browser



```
<html>
  <body>
    Find by email: <input name="email" ng-validate="email"/>
    <ul>
      <li ng-repeat="person in people.$filter(email)">
        {{ person.last | uppercase }},
        {{ person.first }},
      </li>
      <ng:switch on="$location.hashPath">
        <div ng-switch-when="home">Welcome</div>
        <ng:include ng-switch-when="account" src="'account.html'"/>
      </ng:switch>
    </ul>
  </body>
</html>
```

Legend

Scope: Outter most scope which holds services and properties such as '`people`' and '`email`'.

Input Widget: binds to '`email`' in its scope. Changing scope changes the widget and vice versa.

Validator: an input widget may have optional validator to notify user of wrong input.

Directives: instructing the compiler to perform specific actions. In this case a repeater iterates over the list of '`people`' looking for the person with specific email. The DOM element is then replicated to match the number of elements.

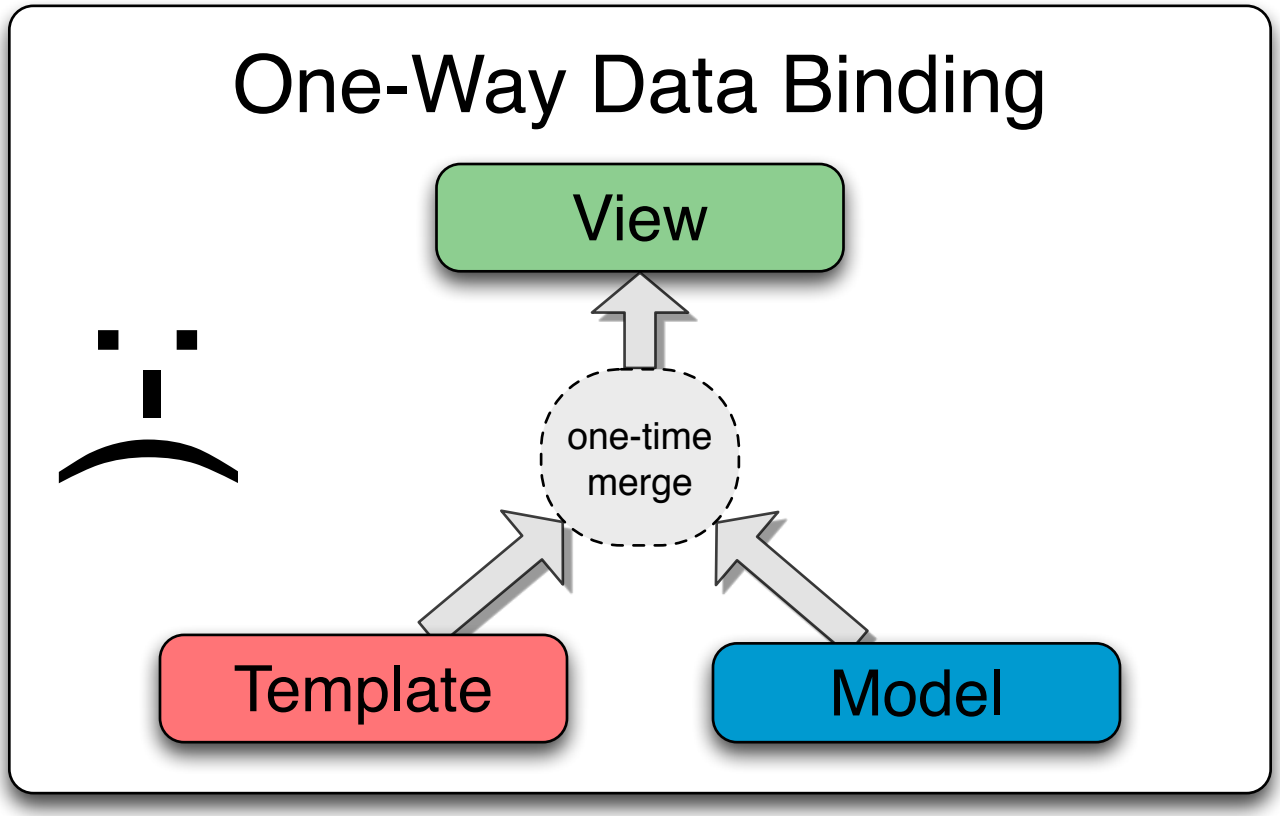
Child Scope: in this case the ng-repeat directive triggers the creation of new scopes one for each item in an iterator expression. It than assigns the 'person' to each scope. The scopes inherit from parent scopes so anything declared at higher scope is still visible.

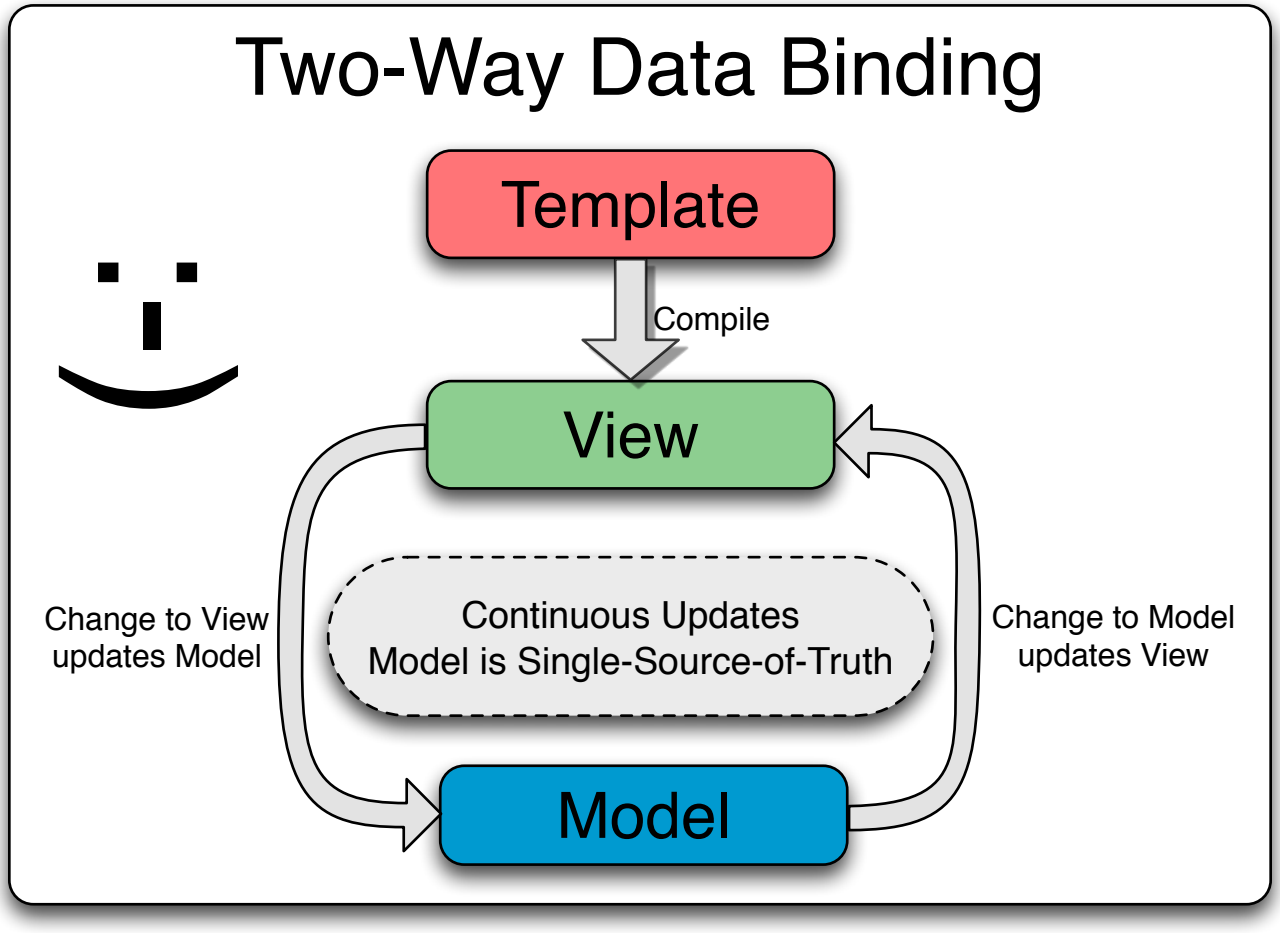
Markup: evaluates expression in the closest scope and inserts it into DOM.

Filter: Markup may include optional filter to transform value before it is displayed.

Widget: Allows the execution of custom code which can transform the DOM.

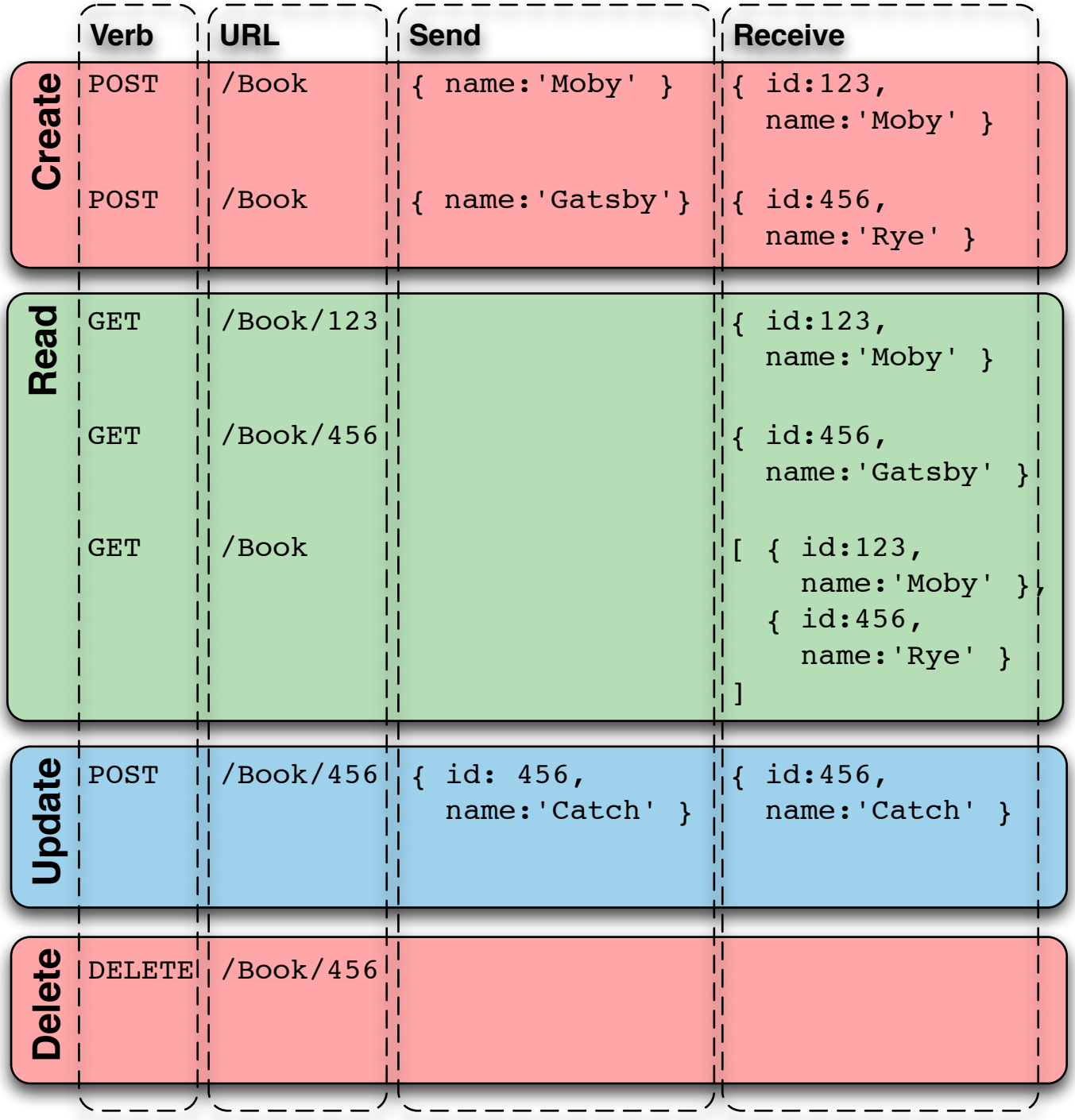
Nested Widgets: widgets can be nested for added expressivness.


One-Way Data Binding
View
one-time
merge
Template
Model

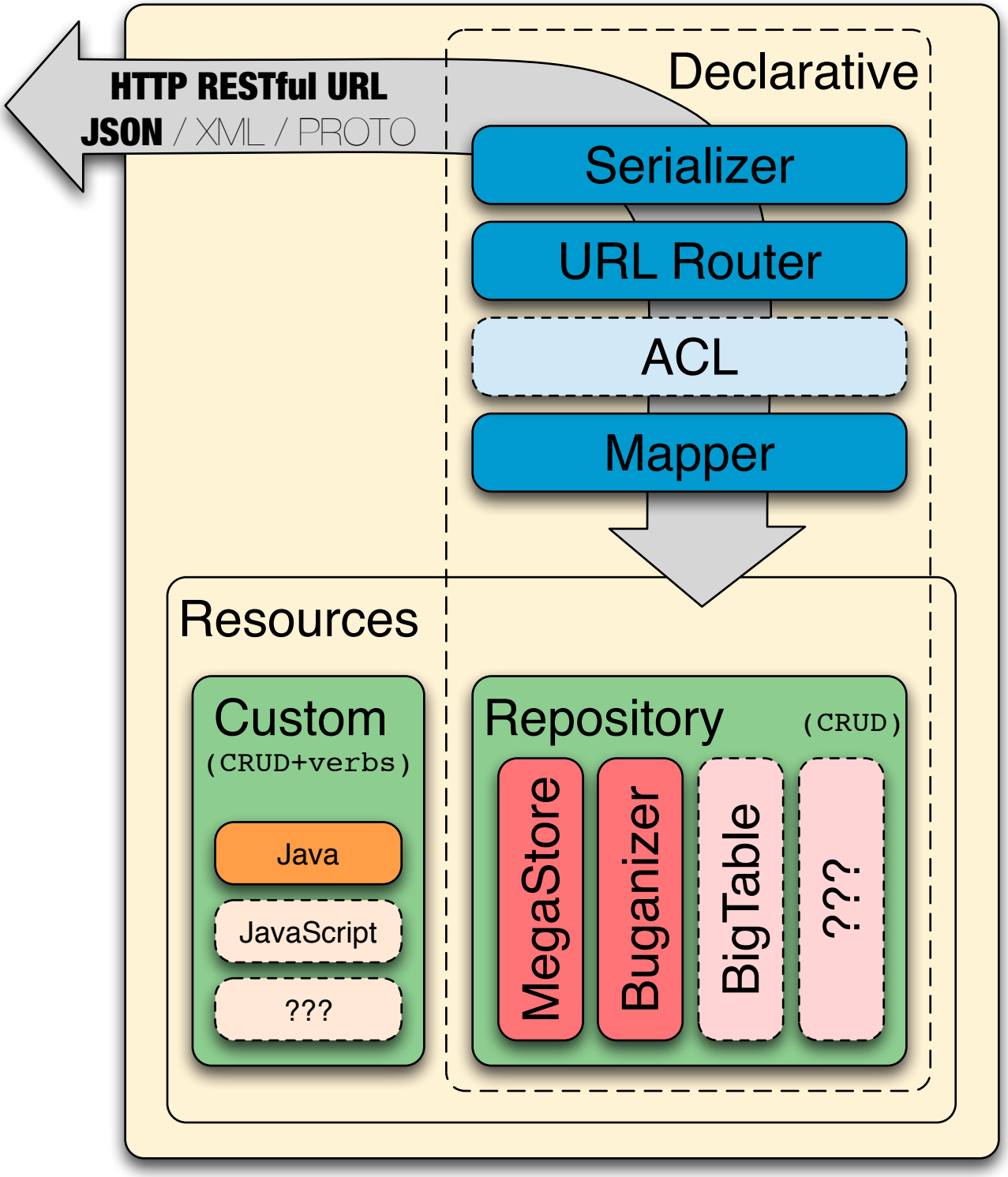

Two-Way Data Binding
Template
Compile
View
Change to View
updates Model
Continuous Updates
Model is Single-Source-of-Truth
Change to Model
updates View
Model

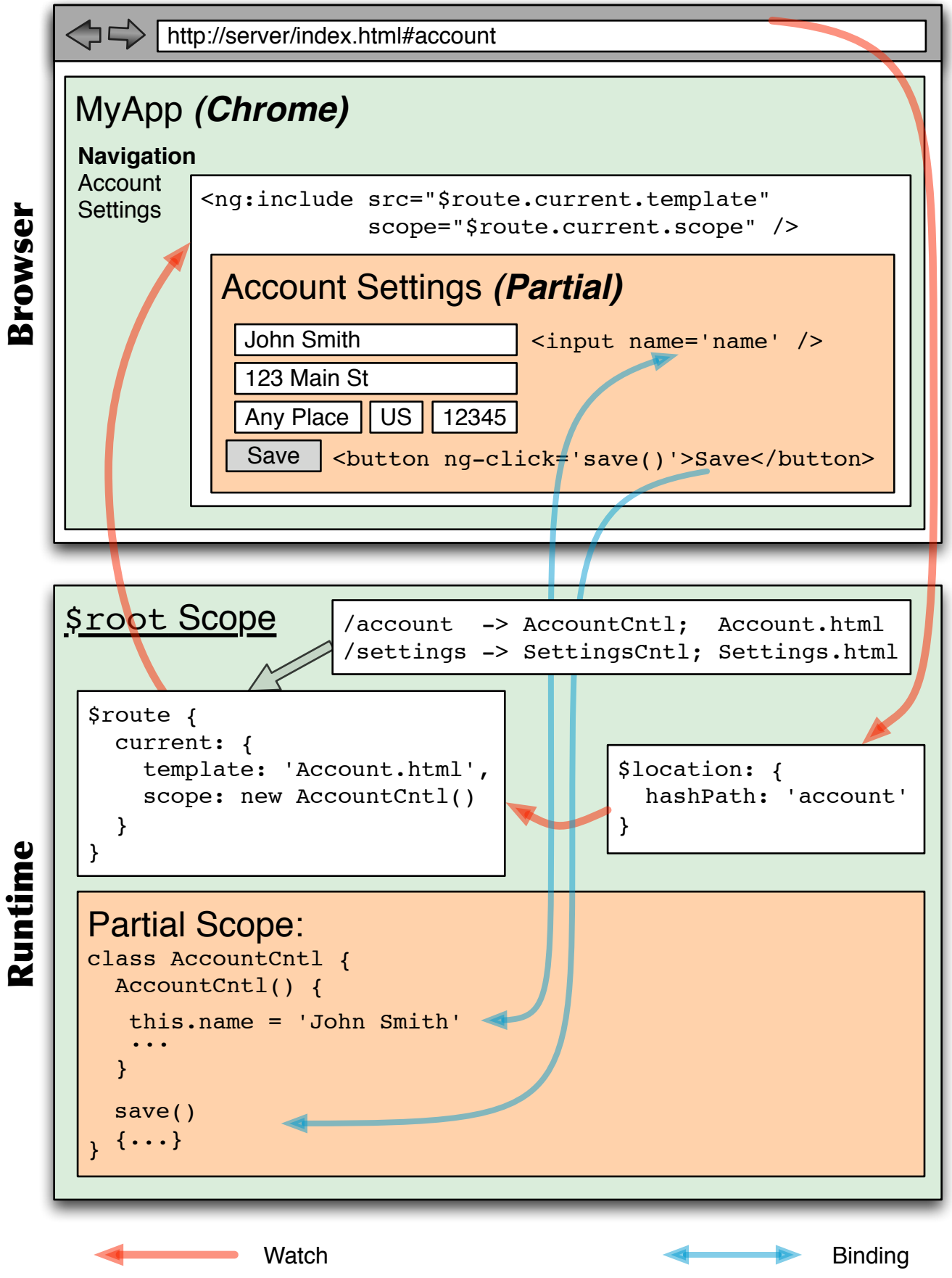
RESTful URLs

`http://server/data/`***`Collection[/ID]`***



RESTy: Declarative Data Storage




Browser
http://server/index.html#account
MyApp (Chrome)
Navigation
Account
Settings
<ng:include src="$route.current.template"
scope="$route.current.scope" />
Account Settings (Partial)
John Smith
<input name='name' />
123 Main St
Any Place
US
12345
Save
<button ng-click='save()'>Save</button>
Runtime
$root Scope
/account -> AccountCntl; Account.html
/settings -> SettingsCntl; Settings.html
$route {
current: {
template: 'Account.html',
scope: new AccountCntl()
}
}
$location: {
hashPath: 'account'
}
Partial Scope:
class AccountCntl {
AccountCntl() {
this.name = 'John Smith'
...
}
save()
} {...}
Watch
Binding