Develop Guide - 1

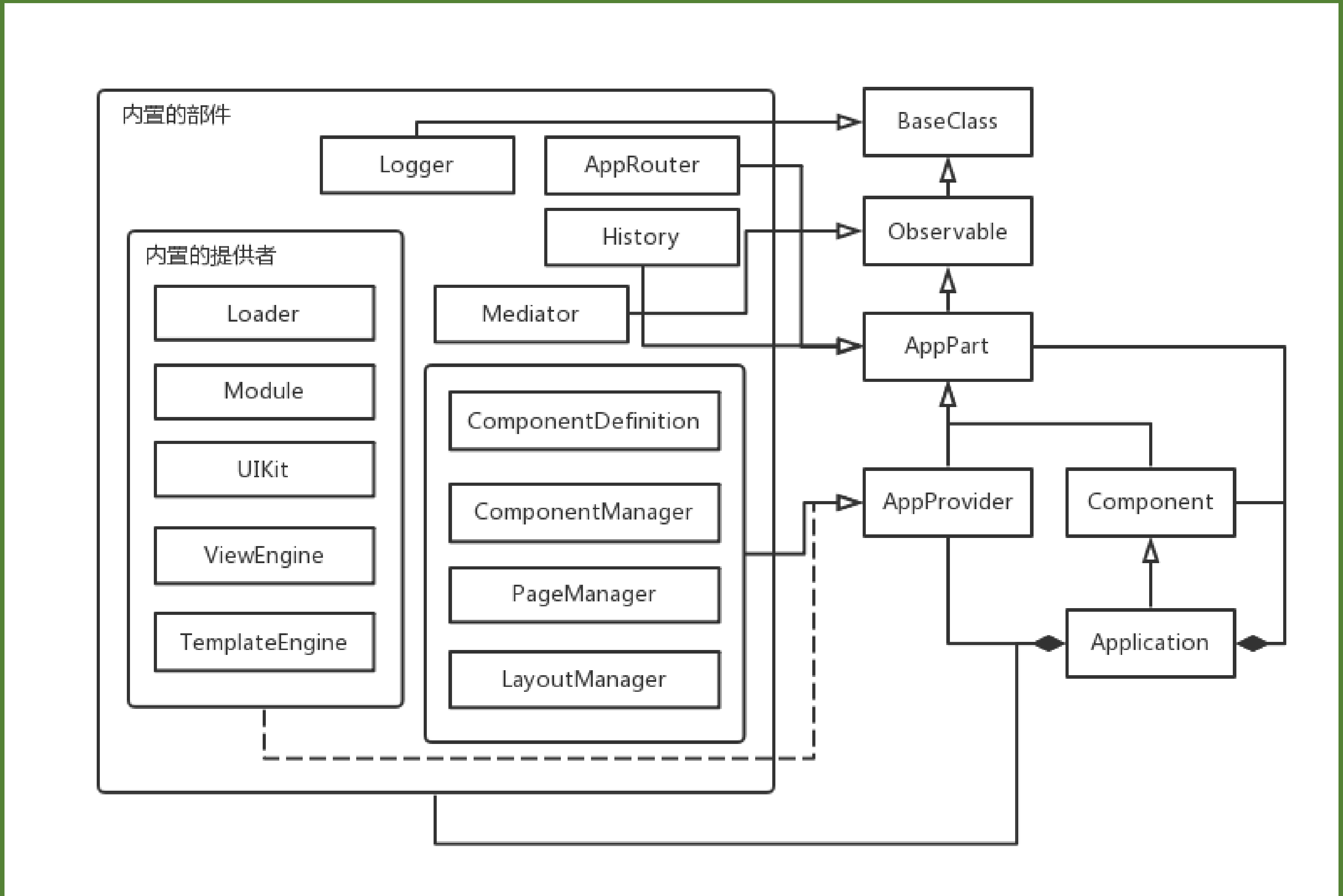
(use veronica 2)

Contents:

- Architecture
  - Store
  - Component

Architecture

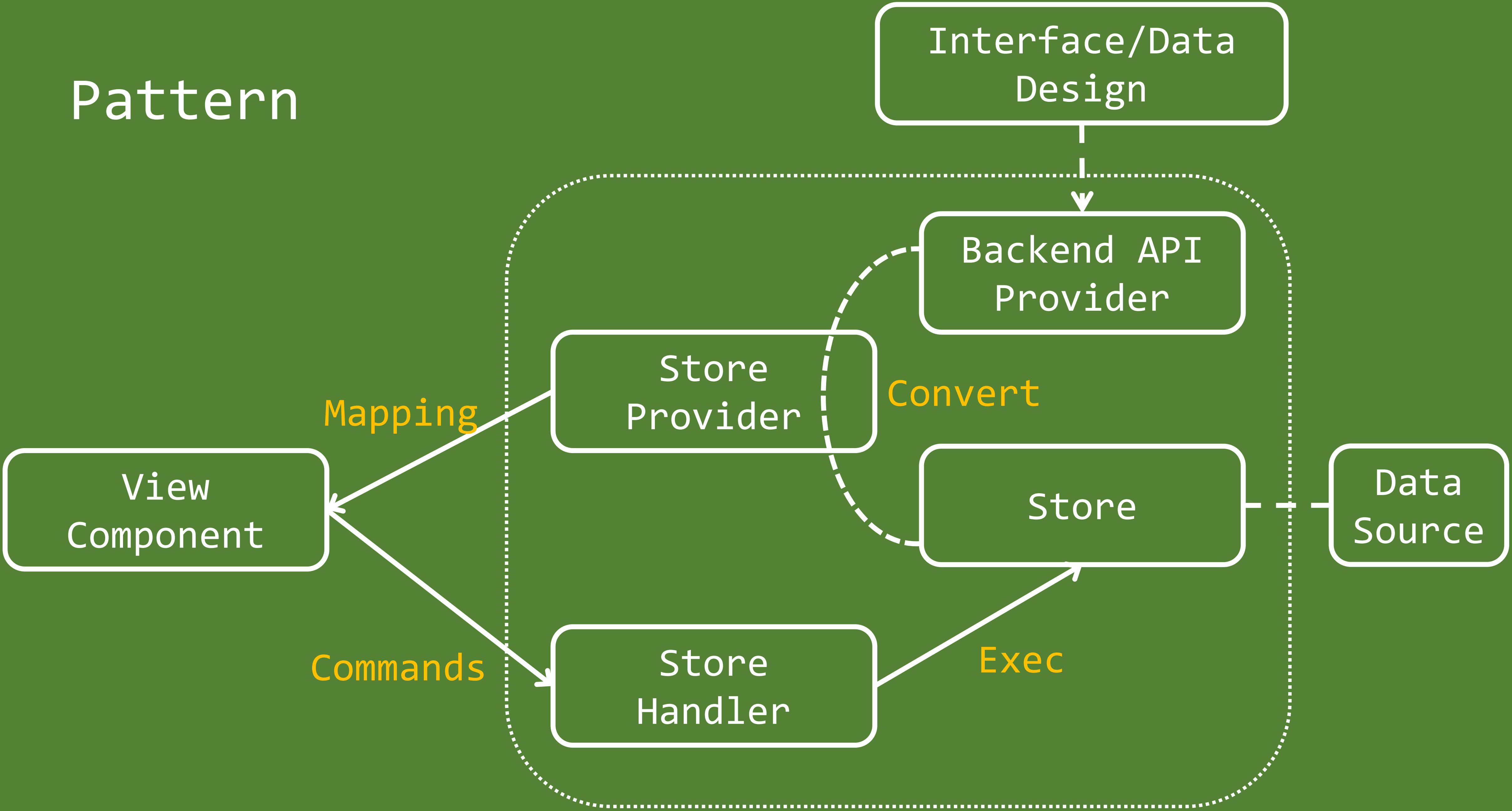
Pattern


内置的部件
Logger
AppRouter
History
Mediator
内置的提供者
Loader
Module
UIKit
ViewEngine
TemplateEngine
ComponentDefinition
ComponentManager
PageManager
LayoutManager
BaseClass
Observable
AppPart
AppProvider
Component
Application

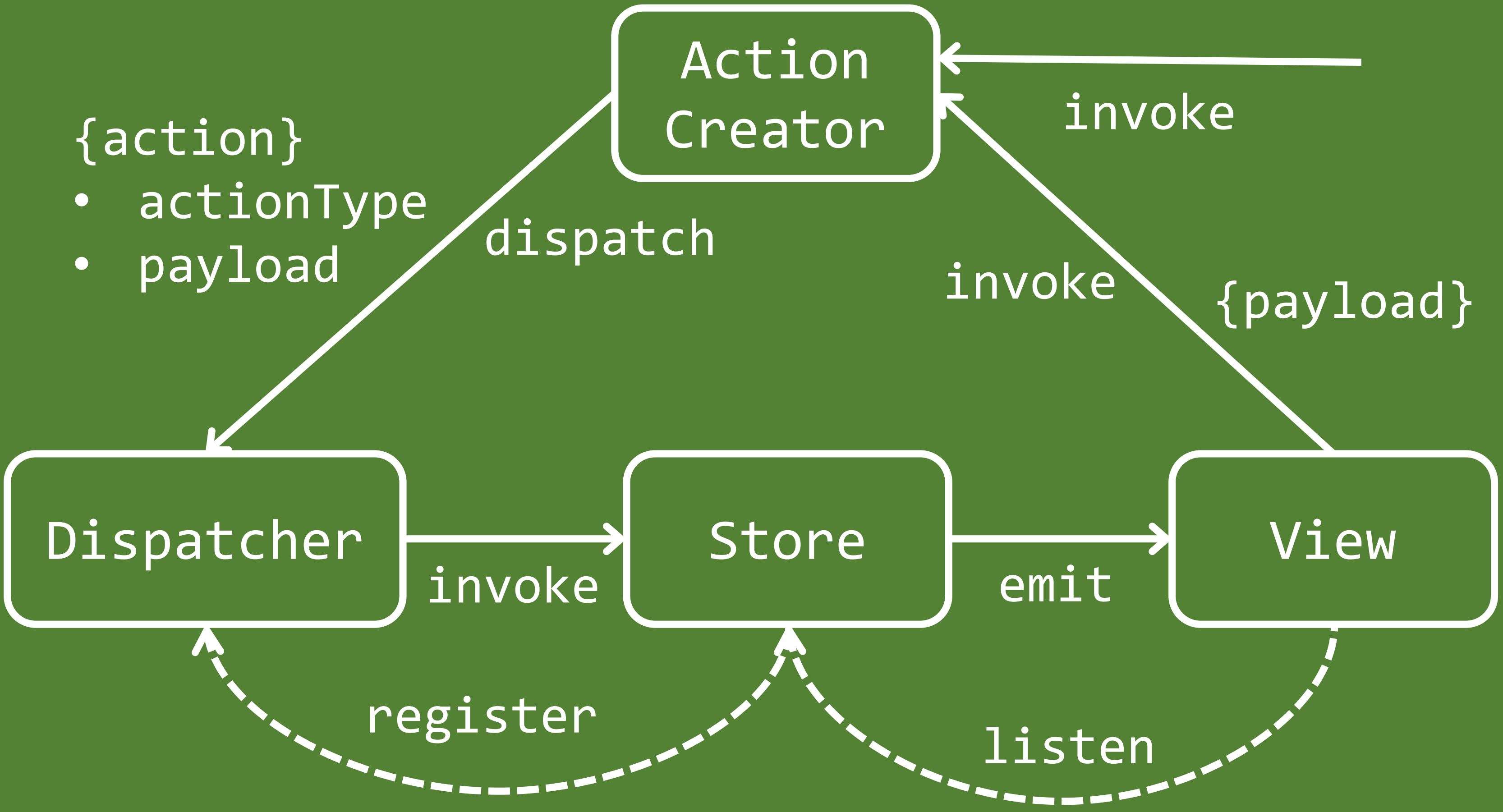
Store

architecture

Pattern


Interface/Data Design
Backend API Provider
Store Provider
Convert
Store
Data Source
Mapping
View Component
Commands
Store Handler
Exec

Analogy: Flux



Action Creator

Dispatcher

Store

View

{action}
- actionType
- payload

dispatch

invoke

invoke

{payload}

invoke

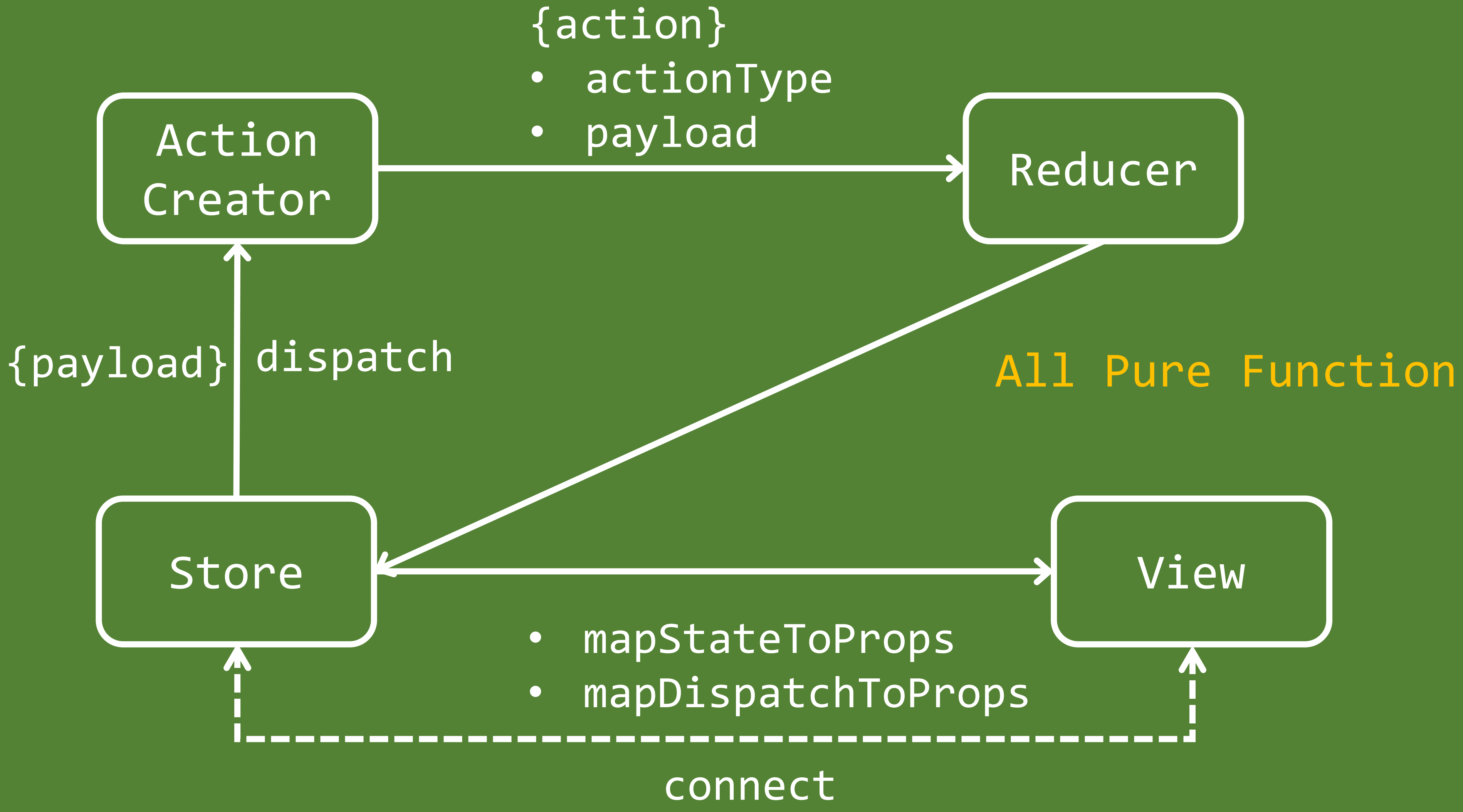
invoke

emit

register

listen

Analogy: Redux

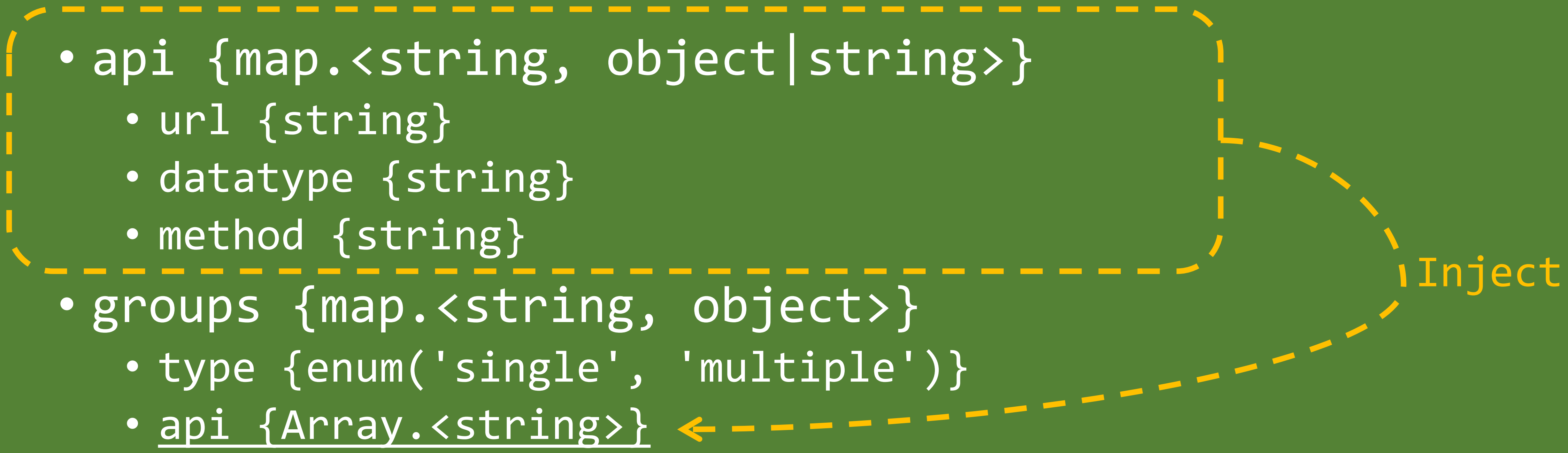


Class Inherit

- Store
  - single (BackendApiDataObject)
    -> kendo.data.ObservableObject
  - multiple (BackendApiDataSource)
    -> kendo.data.HierarchicalDataSource
      -> kendo.data.DataSource

mixin BackendApiDataSource

API: Backend API



- api {map.<string, object|string>}
    - url {string}
    - datatype {string}
    - method {string}
- groups {map.<string, object>}
    - type {enum('single', 'multiple')}
    - api {Array.<string>}
    - resuable {boolean}
    - options {DataSourceOptions}

Inject

API: DataSource Options

- schema
  - type
  - data ← parse response data
  - parse
- transport
  - parameterMap ← parse request data
  - [api]
    - [jQuery Ajax Options]

} ← remote

- data ← local

Component

Architecture

Pattern

```
(BackendApi | Local Db | Runtime Store | ...)

(data) => view

(Browser | Windows | iOS | Android | ...)
```

LifeCycle

1. create -> e:created, hook: created
   - observe
2. render -> e:rendered, hook: rendered
   1. compile -> e:compiled
   2. render
   3. mount -> e:mounted
3. bind -> e: bound, hook: bound
4. destroy -> hook: destroyed

Analogy: React

- options -> props
- model -> state

Render

- [pre =>] compile => element => mount => after

↑ Generate Template

↑ Listen DOM Events

↑ init UI Widget

Development-time why?

Pukit(pug kit) vs JSX

```
+editorFor('name', model)
+panelbar.primary
```

```
{
  id: 'id',
  fields: {
    name: {
      displayName: 'Name',
      type: 'string',
      validation: {
```

<=>

```
<InputNumber min={1}
max={10} defaultValue={3}
onChange={onChange} />
<Card title="Card title">
    <p>Card content</p>
</Card>
```

UI Widget: keboacy

- kendo-ui-core + bootstrap
- Data-Driven
- Widgets
  - DataTable
  - Tree
  - Flowchart
  - Signature
  - Validation
  - ...

Example: Keboacy

```
<div data-role="tree" data-bind="source: treeStore"></div>
```

```
var vm = {
  treeStore: ...
}
kendo.bind($('body'), vm)
```

Events

- this.listen('cmp:this render')
- this.listen('dom:click .selector')
- this.listen('bus:msg')

Q & A