

JavaScript- фреймворки: должен остаться только один

Сергей Аверин
Acronis

 HighLoad⁺⁺



Про что доклад

Как выбирали новый веб-фреймворк

1. Немного о компании
2. Бекграунд
3. Задача
4. Исследование существующего кода
5. Выбор на что смотреть
6. Техническая оценка вариантов
7. Переделка одного из вариантов «под себя»
8. Сравнение пилотных проектов
9. Оценка затрат на внедрение



Про компанию

Масштаб

- **5 000 000** пользователей
- **500 000** из них — корпоративные
- **700** сотрудников в **17** разных офисах
- Выпускаем много разного софта:
 - коробочный под Windows,
 - корпоративный с веб-интерфейсами,
 - cloud-продукты с веб-интерфейсом.

Веб

Все отделы делают веб-часть по-разному



Проблема

Проблема

- Много разных технологий для веб-части
- Фронтенд пишут не только JS-разработчики
- Нет возможности подключить к работе верстальщика
- Качество кода сильно отличается
- Текущие технологии устарели

Задача



Курс

- Толстый клиент на JS/HTML/CSS
- Единая технология во всей компании
- Библиотека UI-компонентов
- Возможность работать разработчикам разных уровней
- Код должен быть понятен backend-разработчикам

Оценка

Что имеем?

- Dojo
- Сайт acronis.com — rich-client там не нужен
- Angular 1.x
- RoR+jQuery
- ExtJS 4

Что не так с ExtJS?

- Индексная страница документации содержит:

Components

Components

Ext.ComponentQuery
Ext.Component
Ext.LoadMask
Ext.ProgressBar
Ext.button.Button
Ext.button.Cycle
Ext.button.Split
Ext.picker.Color
Ext.picker.Date
Ext.picker.Time
Ext.flash.Component
Ext.slider.Multi
Ext.slider.Single
Ext.slider.Tip
Ext.tab.Bar
Ext.tab.Tab
Ext.tip.QuickTip
Ext.tip.QuickTipManager
Ext.tip.Tip
Ext.tip.ToolTip
Ext.MessageBox
Ext.window.MessageBox
Ext.window.Window

Tree

Ext.tree.Panel

Form

Ext.form.Basic
Ext.form.field.Base
Ext.form.field.Checkbox
Ext.form.field.ComboBox
Ext.form.field.Date
Ext.form.field.Display
Ext.form.field.Field
Ext.form.field.File
Ext.form.field.FileButton
Ext.form.field.Hidden
Ext.form.field.HtmlEditor
Ext.form.field.Number
Ext.form.field.Picker
Ext.form.field.Radio
Ext.form.field.Spinner
Ext.form.field.Text
Ext.form.field.TextArea
Ext.form.field.Time
Ext.form.field.Trigger
Ext.form.field.VTypes
Ext.form.RadioGroup
Ext.form.CheckboxGroup
Ext.form.FieldAncestor
Ext.form.FieldContainer
Ext.form.FieldSet
Ext.form.Label
Ext.form.Labelable

Charts

Ext.chart.Callout
Ext.chart.Chart
Ext.chart.Highlight
Ext.chart.Label
Ext.chart.Legend
Ext.chart.LegendItem
Ext.chart.Mask
Ext.chart.Navigation
Ext.chart.Tip
Ext.chart.axis.Axis
Ext.chart.axis.Category
Ext.chart.axis.Gauge
Ext.chart.axis.Numeric
Ext.chart.axis.Time
Ext.chart.series.Area
Ext.chart.series.Bar
Ext.chart.series.Cartesian
Ext.chart.series.Column
Ext.chart.series.Gauge
Ext.chart.series.Line
Ext.chart.series.Pie
Ext.chart.series.Radar
Ext.chart.series.Scatter
Ext.chart.series.Series
Ext.chart.theme.Theme

Drag & Drop

395

классов

HIERARCHY

Ext.Base

└─ Ext.AbstractComponent

└─ Ext.Component

└─ Ext.container.AbstractContainer

└─ Ext.container.Container

└─ Ext.panel.AbstractPanel

└─ Ext.panel.Panel

└─ **Ext.form.Panel**

8 уровней наследования

addBodyCls	enableBubble	getSizeModel	nextSibling	setIcon	updateLayout
addChildEls	ensureAttachedToBody	getState	on	setIconCls	updateRecord
addCls	expand ▶	getValues	prevChild	setLoading	visitPostOrder
addClsWithUI	findParentBy	getViewRegion	previousNode	setLocalX	visitPreOrder
addDocked	findParentByType	getWidth	previousSibling	setLocalXY	
addEvents	findPlugin	getX	query	setLocalY	
addListener	fireEvent	getXTYPE	queryBy	setMargin	
addManagedListener	fireEventArgs	getXTypes	queryById	setOverflowXY ▶	
addMembers ▶ STA	focus	getXY	registerFloatingItem	setPagePosition ▶	
addStateEvents	getActiveAnimation	getY	relayEvents	setPosition ▶	
addStatics ▶ STA	getAlignToXY	hasCls	remove	setRegion ▶	
addTool	getAnchorXY	hasInvalidField	removeAll	setRegionWeight	
addUIClsToElement	getBox	hasListener	removeAnchor ▶	setSize	
alignTo ▶	getChildByElement	hasUICls	removeBodyCls	setTitle	
anchorTo ▶	getCollapsed	hide ▶	removeChildEls	setUI	
animate	getComponent	insert	removeCls	setVisible	
applyState	getConstrainVector	insertDocked	removeClsWithUI	setWidth	
bubble ▶	getDockedComponent	is	removeDocked	setX	
calculateConstrainedPosition	getDockedItems	isAncestor	removeListener	setXY	
cascade ▶	getEI	isDescendantOf	removeManagedListener	setY	
center	getForm	isDirty	removeUIClsFromElement	show ▶	
checkChange	getHeader	isDisabled	render	showAt ▶	
child	getHeight	isDraggable	resumeEvent	showBy ▶	
clearListeners	getIid	isDroppable	resumeEvents	startPolling	
clearManagedListeners	getInitialConfig	isFloating	savePropToState	stopAnimation ▶	
cloneConfig	getInsertPosition	isHidden	savePropsToState	stopPolling	
close	getItemId	isLayoutSuspended	saveState	submit	
collapse ▶	getLayout	isSuspended	scrollBy	suspendEvent	
contains	getLoader	isValid	sequenceFx ▶	suspendEvents	
convertCollapseDir	getLocalX	isVisible	setActive	syncFx ▶	
create STA	getLocalXY	isXTYPE	setAutoScroll ▶	toBack ▶	
createAlias STA	getLocalY	load	setBodyStyle	toFront	
disable ▶	getName STA	loadRecord	setBorder	toggleCollapse ▶	
doAutoRender	getOffsetsTo	mask	setBorderRegion	translatePoints	
doComponentLayout ▶	getPlugin	mon	setBox ▶	un	
doConstrain	getPosition	mon	setDisabled	unmark	

класс с 201 методами

КОТИК ДЕЛАЕТ



МЯУ-МЯУ

ПЕСИК ДЕЛАЕТ



ГАВ-ГАВ

ОВЕЧКА ДЕЛАЕТ



БЕ-БЕ-БЕ

**Твои десятки
методов в классе**



не делают нихера


```
class="x-container x-border-item x-box-item x-container-nav-tab x-layout
: 0px; height: 828px; width: 1488px;">
ab" class="x-panel module-layout x-fit-item x-panel-default" style="margin: 0px,
groups-tab-body" class="x-panel-body x-panel-body-default x-border-layout-ct x-panel-bo
er-trbl" style="width: 1488px; height: 828px; left: 0px; top: 0px;">
div id="panel-1018" class="x-panel nav-tree-panel x-border-item x-box-item x-panel-default" sty
ight: auto; left: 0px; top: 0px; margin: 0px; height: 828px;">
▼ <div id="panel-1018_header" class="x-panel-header x-header x-header-horizontal x-docked x-unse
header-default x-horizontal x-panel-header-horizontal x-panel-header-default-horizontal x-top x-
x-panel-header-default-top x-horizontal-noborder x-panel-header-horizontal-noborder x-panel-head
horizontal-noborder x-docked-top x-panel-header-docked-top x-panel-header-default-docked-top
x-noborder-trl" style="right: auto; left: 0px; top: 0px; width: 400px;">
▼ <div id="panel-1018_header-body" class="x-header-body x-panel-header-body x-panel-header-body-d
header-body-horizontal x-panel-header-body-default-horizontal x-panel-header-body-top x-panel-he
body-default-top x-panel-header-body-horizontal-noborder x-panel-header-body-default-horizontal-n
x-panel-header-body-docked-top x-panel-header-body-default-docked-top x-box-layout-ct x-panel-hea
body-default-horizontal x-panel-header-body-default-top x-panel-header-body-default-horizontal-nob
x-panel-header-body-default-docked-top" style="width: 358px;">
▼ <div id="panel-1018_header-innerCt" class="x-box-inner " role="presentation" style="width: 358px;
64px;">
▼ <div id="panel-1018_header-targetEl" class="x-box-target" style="width: 358px;">
▼ <div id="panel-1018_header_hd" class="x-component x-header-text-container x-panel-header
text-container x-panel-header-text-container-default x-box-item x-component-
default" unselectable="on" style="right: auto; left: 0px; top: 0px; margin: 0px; width: 2
<span id="panel-1018_header_hd-textEl" class="x-header-text x-panel-header-text x-pa
text-default" unselectable="on">Groups</span>
</div>
▼ <div id="tool-1523" class="x-tool x-box-item x-tool-default x-tool-after-title" style="w
height: 48px; right: auto; left: 310px; top: 8px; margin: 0px;">

</div>
</div>
</div>
</div>
▼ <div id="panel-1018-body" class="x-panel-body x-panel-body-default x-border-layout-ct x-panel-body-
x-noborder-trbl" style="width: 400px; height: 763px; left: 0px; top: 65px;">
▼ <div id="groups-tree" class="x-panel x-grid-no-row-lines x-grid-header-hidden x-border-item x
nav-tree x-tree-panel x-tree-arrows x-grid" style="right: auto; left: 0px; top: 0px; margin:
height: 699px;">
▼ <div id="headercontainer-1013" class="x-grid-header-ct x-grid-header-ct-hidden x-docke
ct-default x-docked-top x-grid-header-ct-docked-top x-grid-header-ct-default-docked-t
x-noborder-trl" style="height: 0px; right: auto; left: 0px; top: 0px; width: 400px;">
▼ <div id="headercontainer-1013-innerCt" class="x-box-inner " role="presentation"
height: 0px;">
▼ <div id="headercontainer-1013-targetEl" class="x-box-target" style="wid
▼ <div id="treecolumn-1014" class="x-column-header x-column-header-
header-default x-unselectable x-column-header-first x-column-hea
left: 0px; top: 0px; margin: 0px; width: 400px; height: 0px;">
▼ <div id="treecolumn-1014-titleEl" class="x-column-head
<span id="treecolumn-1014-textEl" class="x-col
</div>
```

```

▼ <div id="container-1502" class="x-container x-border-item x-box-item x-container-nav-t
192px; top: 0px; margin: 0px; height: 828px; width: 1488px;">
  ▼ <div id="groups-tab" class="x-panel module-layout x-fit-item x-panel-default" styl
828px;">
    ▼ <div id="groups-tab-body" class="x-panel-body x-panel-body-default x-border-la
x-noborder-trbl" style="width: 1488px; height: 828px; left: 0px; top: 0px;">
      ▼ <div id="panel-1018" class="x-panel nav-tree-panel x-border-item x-box-iter
right: auto; left: 0px; top: 0px; margin: 0px; height: 828px;">
        ▼ <div id="panel-1018_header" class="x-panel-header x-header x-header-hor
header-default x-horizontal x-panel-header-horizontal x-panel-header-de
x-panel-header-default-top x-horizontal-noborder x-panel-header-horizor
horizontal-noborder x-docked-top x-panel-header-docked-top x-panel-head
x-noborder-trl" style="right: auto; left: 0px; top: 0px; width: 400px;"
          ▼ <div id="panel-1018_header-body" class="x-header-body x-panel-heade
header-body-horizontal x-panel-header-body-default-horizontal x-pan
body-default-top x-panel-header-body-horizontal-noborder x-panel-he
x-panel-header-body-docked-top x-panel-header-body-default-docked-t
body-default-horizontal x-panel-header-body-default-top x-panel-hea
x-panel-header-body-default-docked-top" style="width: 358px;"
            ▼ <div id="panel-1018_header-innerCt" class="x-box-inner " role="
64px;">
              ▼ <div id="panel-1018_header-targetEl" class="x-box-target" s
                ▼ <div id="panel-1018_header_hd" class="x-component x-heac
text-container x-panel-header-text-container-default x-l
default" unselectable="on" style="right: auto; left: 0px
                  <span id="panel-1018_header_hd-textEl" class="x-heac
text-default" unselectable="on">Groups </span>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</pre>

```

~1% DOM-дерева главной

Кастомный UI компонент

Это станет CSS

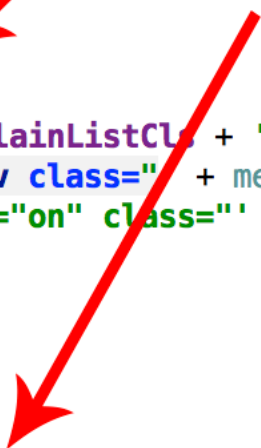


```
defaultListConfig: {  
  loadingHeight: 70,  
  minWidth: 70,  
  maxHeight: 300,  
  shadow: 'sides'  
},
```

А это вообще какая-то адская
мешанина из HTML, кода
шаблонизатора и JS-кода.




```
me.listTpl = ['<ul class="" + Ext.plainListCls + ""><tpl for=".">',  
  '<tpl if="xindex == xcount"><div class="" + me.baseCls + '-separator ' + Ext.baseCls + '-separator',  
  '<li role="option" unselectable="on" class="" + Ext.baseCSSPrefix + 'boundlist-item',  
  '</tpl></ul>'];
```




```
if (!me.displayTpl) {  
  me.displayTpl = new Ext.XTemplate(  
    '<tpl for=".">' +  
    '{[typeof values === "string" ? values : values["' + me.displayField + '"]]}',  
    '<tpl if="xindex < xcount">' + me.delimiter + '</tpl>' +  
    '</tpl>'
```

layouting

▼ Call Stack		<input type="checkbox"/> Async
Ext.layout.ContextItem#setProp	ContextItem.js:1475	
Ext.layout.ContextItem#setHeight	ContextItem.js:1541	
	Box.js:1098	
Ext.layout.container.Box#publishInnerCtSize		
Ext.layout.container.Box#calculate	Box.js:529	
Ext.layout.Context#runLayout	Context.js:1154	
Ext.layout.Context#runCycle	Context.js:1123	
Ext.layout.Context#run	Context.js:1054	
	AbstractComponent.js:111	
Ext.AbstractComponent.flushLayouts		
	AbstractComponent.js:174	
Ext.AbstractComponent.updateLayout		
	AbstractComponent.js:3319	
Ext.AbstractComponent#updateLayout		
Ext.layout.Layout#onContentChange	Layout.js:491	
	AbstractComponent.js:3314	
Ext.AbstractComponent#updateLayout		
	AbstractComponent.js:1454	
Ext.AbstractComponent#onShow		
Ext.Base#callParent	ext-dev.js:6952	
Ext.Component#onShow	Component.js:980	
Ext.Base#callParent	ext-dev.js:6952	
Ext.panel.Panel#onShow	Panel.js:1267	
Ext.Component#show	Component.js:938	
Acronis.navigation.set.done	Backup.js:1251	

test 

WHAT TO BACK UP	Files/folders
ITEMS TO BACK UP	C:\Documents and Settings\ C:\Recovery\
WHERE TO BACK UP	C:\Recovery\
SCHEDULE	Monday to Friday at 11:00 PM (Always full)
HOW LONG TO KEEP	Daily: 7 days Weekly: 4 weeks Monthly: 6 months

 Backup failed

The disk is full.

```
toolbar-1453" class="x-toolbar bottom-bar x-border-item x-toolbar-default x-box-layout-ct" style="border-width: 0px; height: 64px; right: auto; left: 0px; top: 765px; margin: 0px; width: 1088px;"/>

```

layouting

Apr 07, 2015, 11:04 PM [Download log](#) [Details](#)

RUN NOW

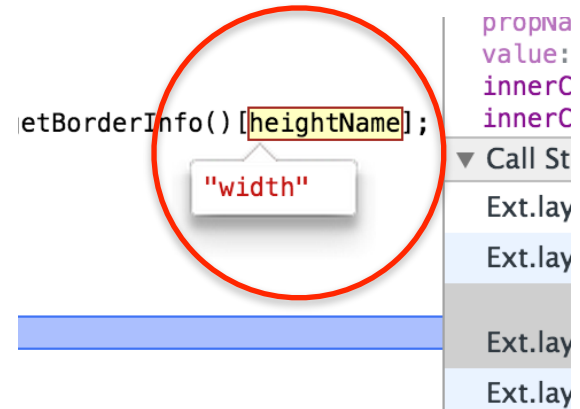
deep in layouting code...

```
if (align.stretch) {
    innerCtHeight = height;
} else {
    innerCtHeight = plan.maxSize
        + padding[names.beforeY]
        + padding[names.afterY]
        + innerCtContext.getBorderInfo()[heightName];

    if (!ownerContext.perpendicularSizeMode.shrinkWrap
        && (align.center || align.bottom)) {
        innerCtHeight = Math.max(height, innerCtHeight);
    }
}
```

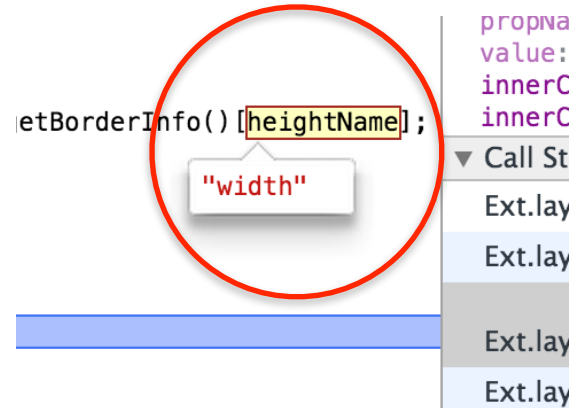
```
innerCtContext[names.setWidth](innerCtWidth);
innerCtContext[names.setHeight](innerCtHeight);
```

```
// If unable to publish both dimensions, this layout needs to run again
if (isNaN(innerCtWidth + innerCtHeight)) {
    me.done = false;
}
```

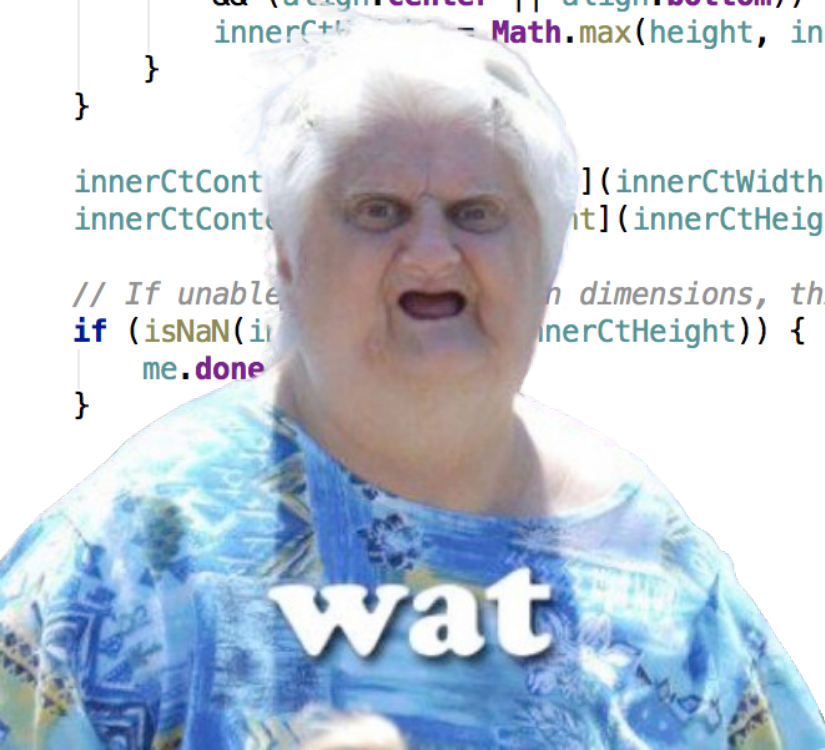


deep in layouting code...

```
if (align.stretch) {  
    innerCtHeight = height;  
} else {  
    innerCtHeight = plan.maxSize  
        + padding[names.beforeY]  
        + padding[names.afterY]  
        + innerCtContext.getBorderInfo()[heightName];  
  
    if (!ownerContext.perpendicularSizeMode.shrinkWrap  
        && (align.center || align.bottom)) {  
        innerCtHeight = Math.max(height, innerCtHeight);  
    }  
}
```



```
innerCtContext.setWidth(innerCtWidth);  
innerCtContext.setHeight(innerCtHeight);  
  
// If unable to determine dimensions, this layout needs to run again  
if (isNaN(innerCtWidth) || isNaN(innerCtHeight)) {  
    me.done  
}
```



**Вы еще помните,
что эта секция называется
«Производительность фронтенда»?**

URL	Status	Domain	Size	Remote IP	Timeline
▶ GET baas.acrc	200 OK	baas.acronis.com	1.9 KB	85.25.24.49:443	54ms
▶ GET Acronis-	200 OK	baas.acronis.com	58 B	85.25.24.49:443	54ms
▶ GET app.js?v=	200 OK	baas.acronis.com	1.7 MB	85.25.24.49:443	12.7s
▶ GET base.css:	200 OK	baas.acronis.com	7.8 KB	85.25.24.49:443	57ms
▶ GET css?fami	200 OK	fonts.googleapis.com	362 B	64.233.163.95:443	31ms
▶ GET Acronis-	200 OK	baas.acronis.com	306.3 KB	85.25.24.49:443	339ms
▶ GET Acronis-	200 OK	baas.acronis.com	5.5 KB	85.25.24.49:443	367ms
▶ GET ui?_dc=1	200 OK	baas.acronis.com	14 B	85.25.24.49:443	55ms
▶ GET cloud-ge	200 OK	baas.acronis.com	5.3 KB	85.25.24.49:443	111ms
▶ GET status_m	200 OK	baas.acronis.com	1.4 KB	85.25.24.49:443	55ms
▶ GET state_ma	200 OK	baas.acronis.com	1.2 KB	85.25.24.49:443	479ms
11 requests			2.0 MB	16.07s (onload: 16.07s)	

Рафик, где мой трафик?

**Ладно с фреймворком понятно, а
само приложение?**

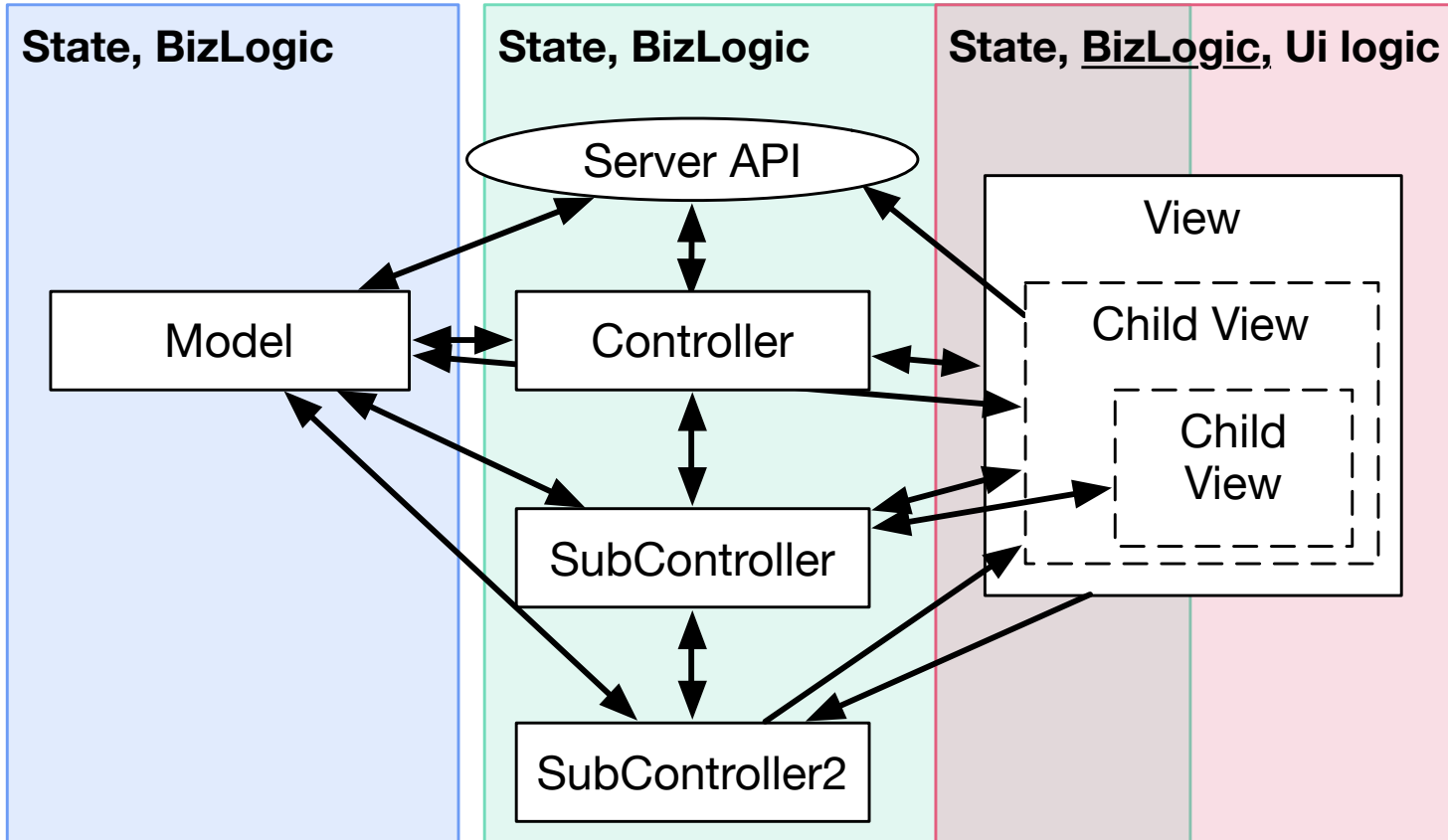
Полезли в код приложения

- Мало комментариев
- Жесткая связность
- Нет границы между Controller и View

M

C

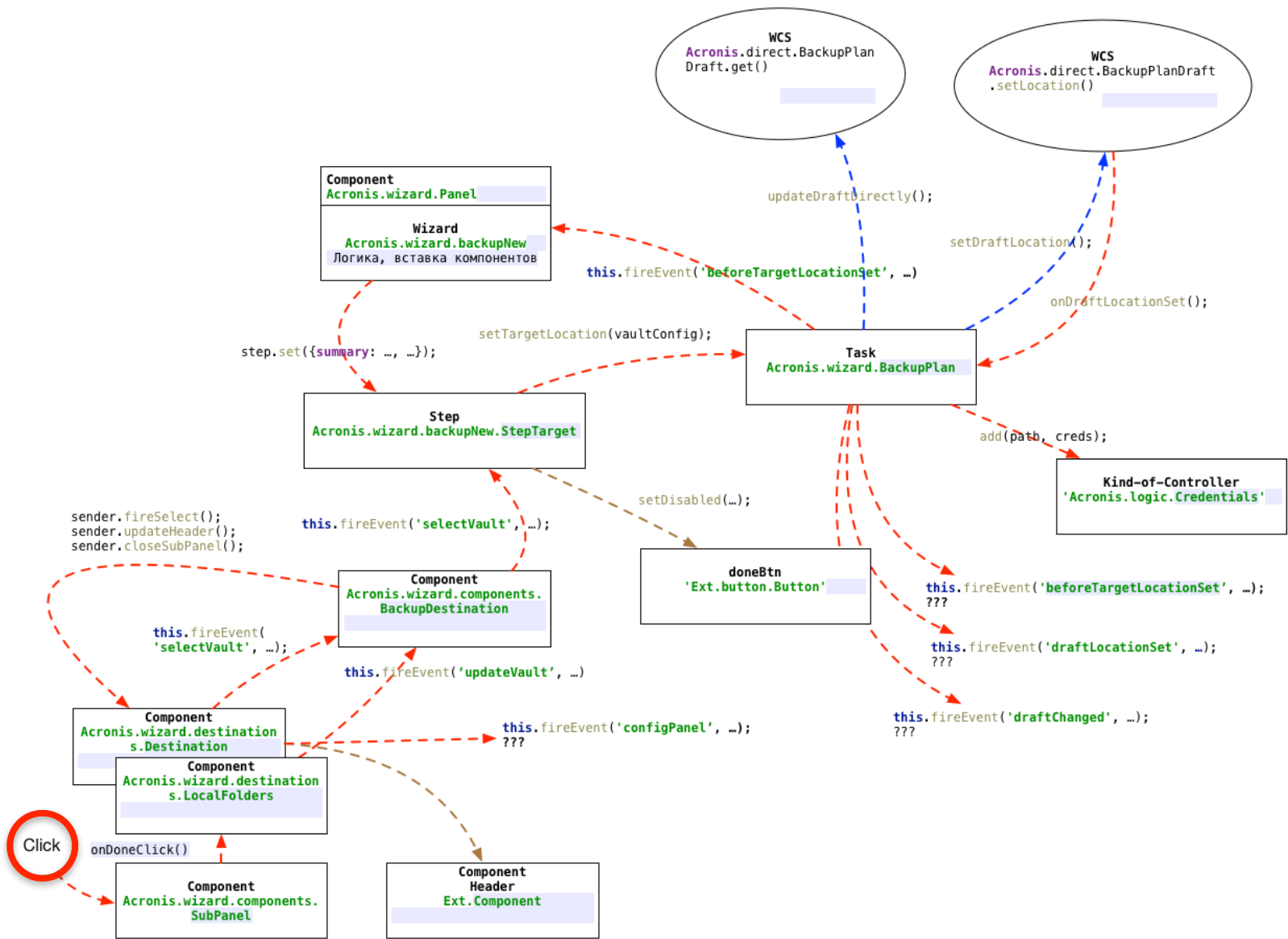
V



M+CV

Полезли в код приложения

- Мало комментариев
- Жесткая связность
- Нет границы между Controller и View
- **Publish/Subscribe** – вроде как правильный паттерн, но не работает.




8 СОБЫТИЙ, ~18 ВЫЗОВОВ



Вопрос на засыпку

Чего обычно **нет** в коде приложений с жесткой связностью и плохим разграничением зон ответственности?

- ▼ test
 - ▼ 
 - ▼ common
 - ▼ logic
 - Utils.compareArrays.js
 - Utils.compareObjects.js
 - Utils.isSubSet.js
 - ▼ controller
 - Backup.js
- ▼ helpers
 - Controller.js



Полезли в код приложения

Выводы...

Полезли в код приложения



Выводы...

Настоящие выводы

- **Очень сложный фреймворк**
- **Запутанный получившийся код**
- **Мегабайты кода**
- **Нельзя подключить обычных верстальщиков**
- **Виноват ли фреймворк? – частично**

Куда направить усилия?

- нужен проще UI слой
- менее связанная архитектура
- четкое разделение зон ответственности (языков, технологий)
- больше границ и правил для программистов
- и код, в котором просто разобраться среднестатистическому разработчику.

Второй вывод

«производительность фронтенда»:

Улучшаем производительность фронтенд-разработчиков
(технологии + стандарты)



Получаем производительность фронтенд-продуктов

Процесс выбора

*«Хорошие художники копируют,
великие художники воруют.»*

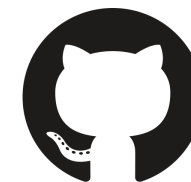
Пабло Пикассо

...что нам подскажет интернет?

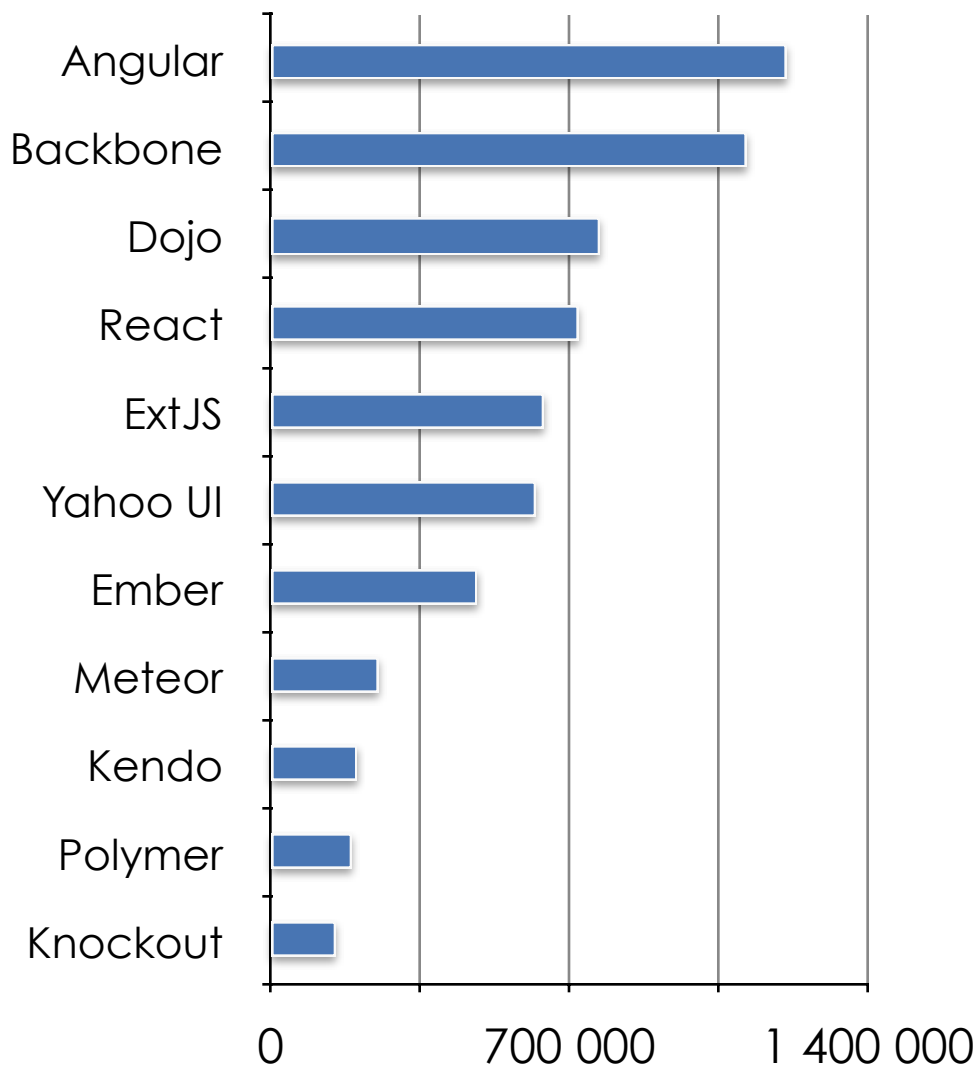
Что нам подскажет интернет?

Angular, backbone, meteor, **Ember**, polymer, Aurelia, **React**, Vue, Mercury, Dojo, Knockback.js, CanJS, Mithril, Ampersand, **Knockout**, Flight, TroopJS, Batman, Spine, YUI, ExtJS, Google Web Toolkit, Kendo UI, OpenUI5, Webix, Echo3, Enyo

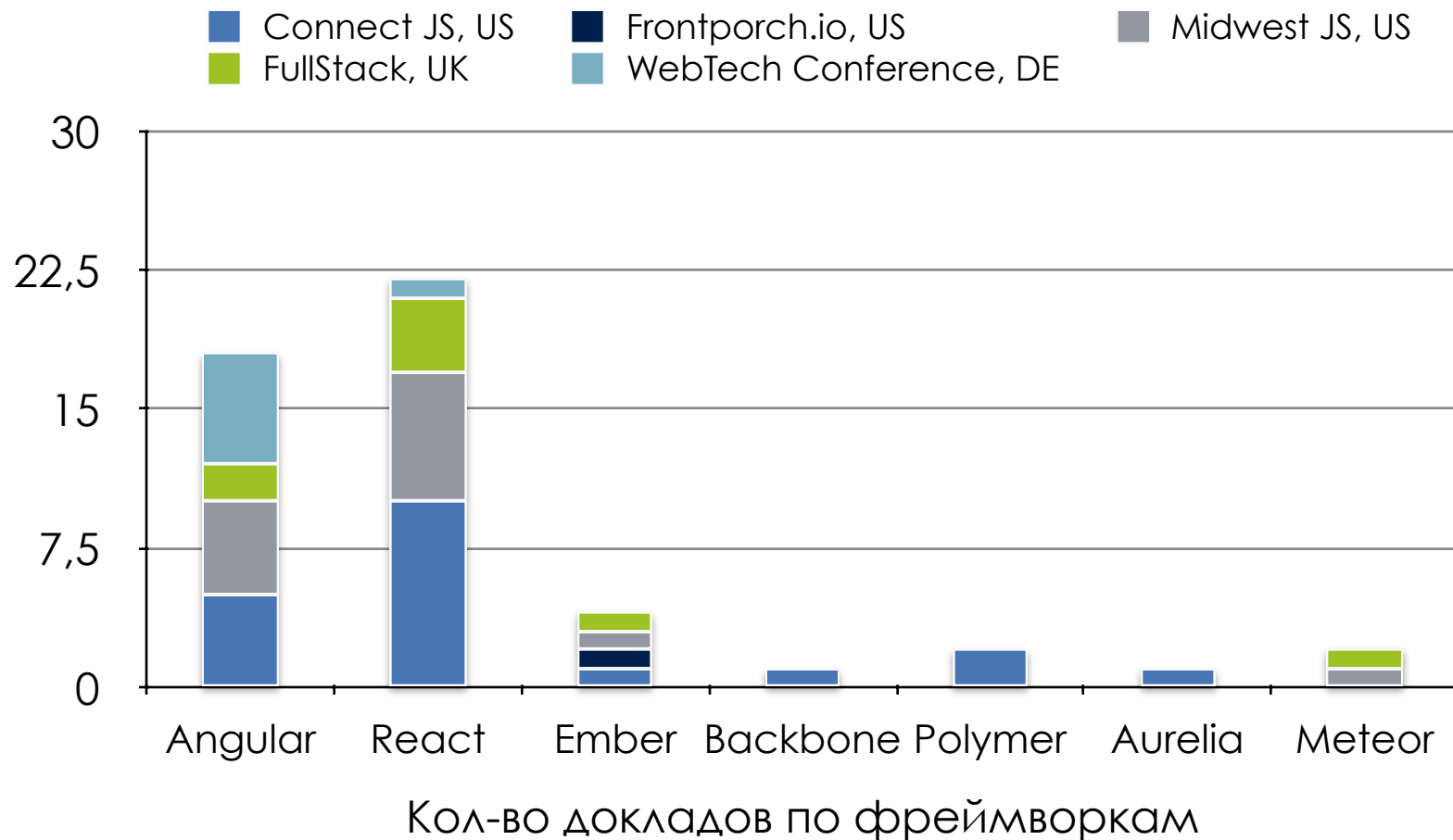
GitHub



Кол-во строк кода



Конференции 2015



Тренды

Angular js

Поисковый запр...

backbon... ×

Поисковый запр...

react js

Поисковый запр...

knockout js

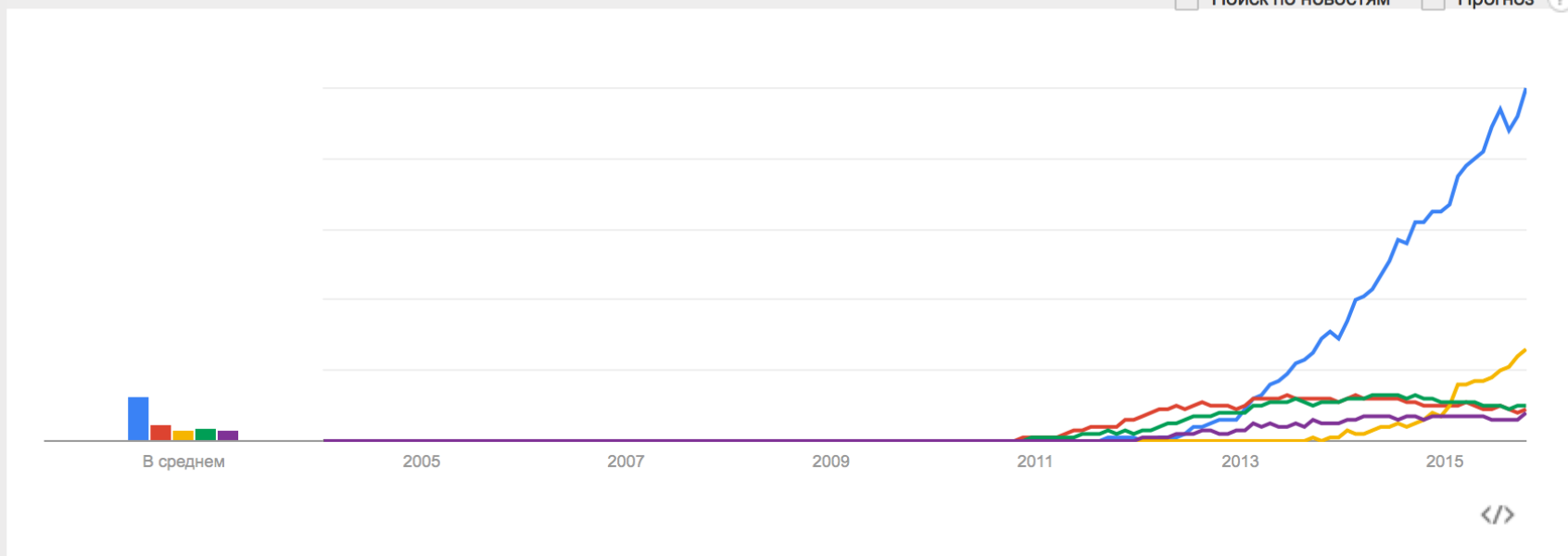
Поисковый запр...

ember js

Поисковый запр...

Динамика популярности ?

Поиск по новостям Прогноз ?



Тренды

Angular js

Поисковый запр...

dojo js

Поисковый запр...

yahoo ui

Поисковый запр...

google w...

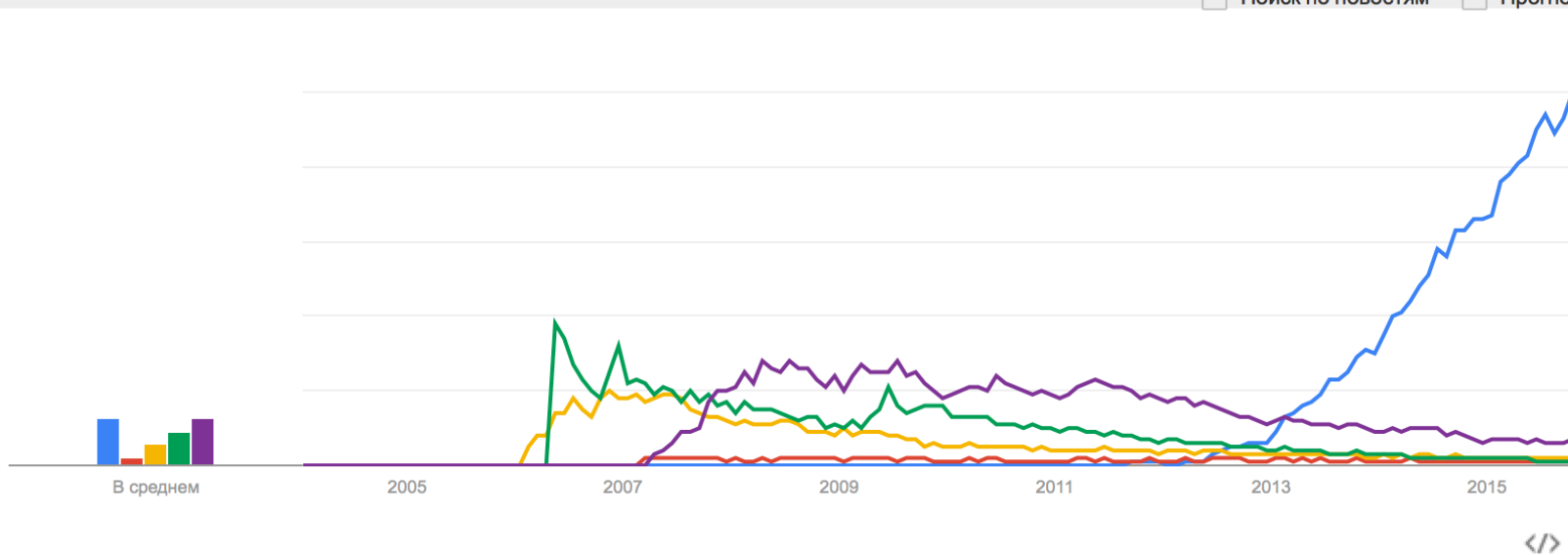
Поисковый запр...

Ext js

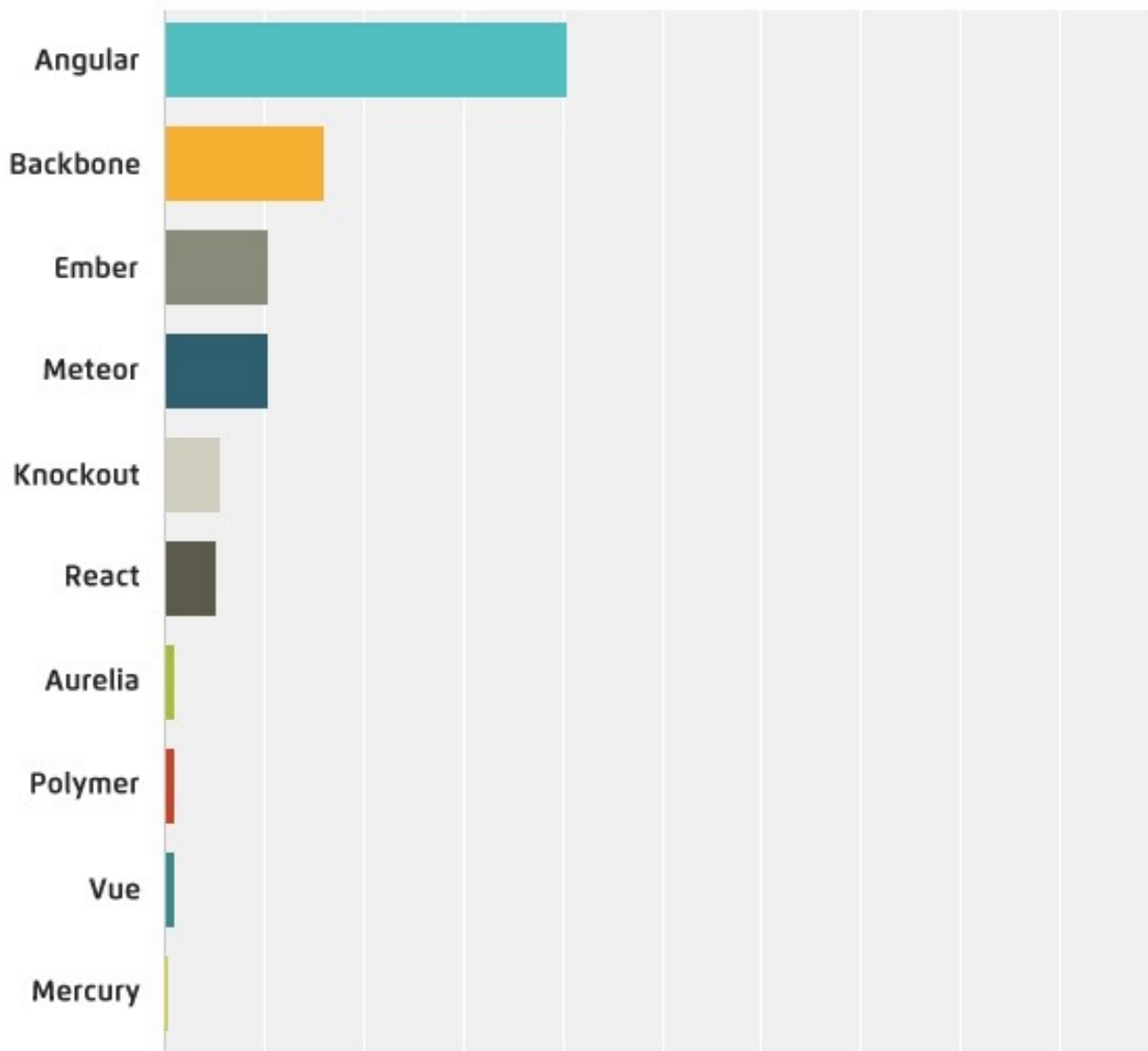
Поисковый запр...

Динамика популярности ?

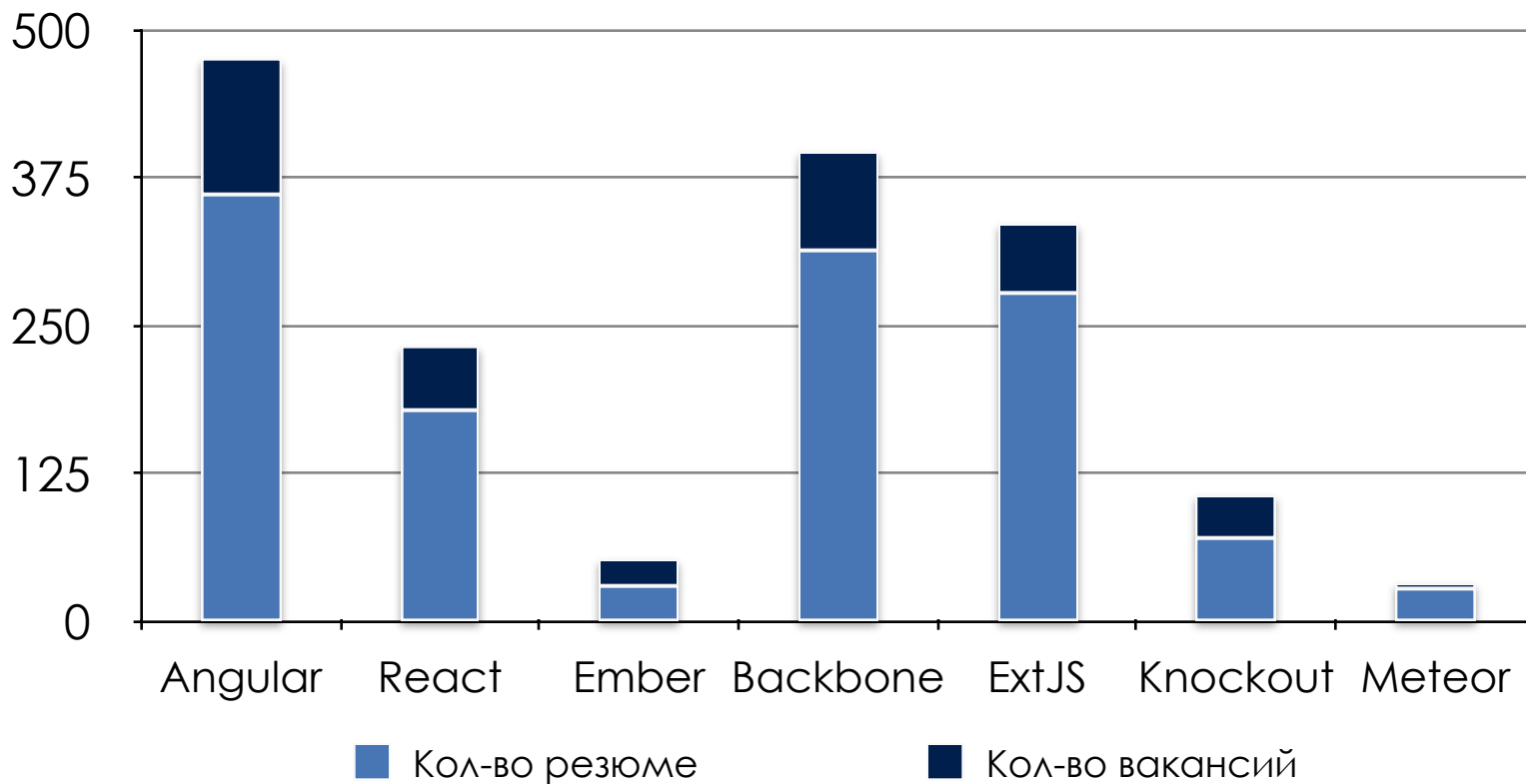
Поиск по новостям Прогноз ?



Codeanywhere (cloud IDE)



Рынок вакансий



Что смотрим:

- AngularJS
- Ember
- Knockout
- Backbone.js + проекты-расширения
- React + Flux
- Dojo
- ExtJS 6 =)

<cut />: что не подошло

- Backbone
- Ember
- Knockout
- Dojo
- ExtJS 6



Версия 1.*

- Хорошая модульность
- Нет единого стиля разработки
- Трудно дебажить
- Многовато «магии»
- Сложно интегрировать с новыми технологиями
- Код будет несовместим с версией 2.*



1 or 2?

Посмотрели 2.0.0-alpha

- *Аж три языка – TypeScript, Javascript и Dart.*
- *Вот это выглядит как то, что надо...*
- *...только шансов на скорый релиз нет.*
- *...и везде в документации написано «предварительно»...
...«может поменяться»...*



Не фреймворк, а UI-библиотека

- структурность
- понятный data flow, изолированность компонент
- нет какого-то магического синтаксиса (кастомных атрибутов, фильтров)
- понятная и простая возможность тюнинга производительности
- и даже серверный рендеринг



Архитектура, но не фреймворк

- + one-way data flow, синхронная обработка
- + Как приложение делится на независимые блоки с помощью денормализации — понятно
- + Единый Event Bus (Dispatcher) и уникальные события — круто.
- Как обеспечивается динамика — непонятно
- Смешение концепций — Store'ы и хранят данные и реализуют бизнес-логику...



- Кода от самого Facebook считай, почти нет.
- Посмотрели реальные фреймворки – fluxxor, DeLorean, ReFlux.js, Este.js:
 - уже лучше, но все еще нет динамики
 - видно общий прогресс стандартизации в виде ES6, прт-модулей, изоморфности и т. д.
 - с разработкой веб-приложений беда... невозможно, например, создать два instance'а одного Store'а, чтобы они не воевали друг с другом.
 - нет интернационализации
 - нет примеров тестов

Попутно нашли...

Typescript

Шанс уменьшить «креативность» разработчиков разных отделов

```
class A {  
    public say() {  
        var abc:string = '123';  
        console.log(abc);  
    }  
}
```

```
var v = new A();  
v.say();
```



```
var A = (function () {  
    function A() {  
    }  
    A.prototype.say = function () {  
        var abc = '123';  
        console.log(abc);  
    };  
    return A;  
})();  
var v = new A();  
v.say();
```


Контракты

```
type ProfileData = {  
  id: number,  
  group: {id: number}  
}
```

```
class API {  
  getProfileData(id: string): ProfileData {...}  
}
```

```
export class ProfileDataResponse extends Action<ProfileData> {  
  protected name = 'ApiClient:ProfileDataResponse';  
}
```

Интерфейсы

```
/**  
 * Just a general Controller  
 */  
export interface IController extends IActionSubscriber {  
    //parent: IController; //must be present in all non-page  
  
    init(params?: Object): void;  
    dispatch(action: Action<any>, is_global?: boolean): void;  
}
```

А также

- Дженерики
- Декораторы
- Составные типы

В итоге?

- Вопросов стало только больше =)
- «Серебряной пули» нет. В этом плане ExtJS «держит удар».
- Хотим фреймворк с Typescript!

Сфокусируемся

Вернемся к задаче

- JS-кодеры должны писать код, верстальщики — делать шаблоны
- Нужен проще UI слой, понятнее архитектура, четкое разделение (языков, технологий), больше границ, правил и стандартов.
- На Typescript
- Не являющийся монолитным все-в-одном

Портируем Flux

- Взяли за основу Flux+React фреймворк Este.js, как наиболее инновационный.
- Плавно переписывали, пока за три захода от него не осталось ничего, кроме конфига сборщика.

Трудности

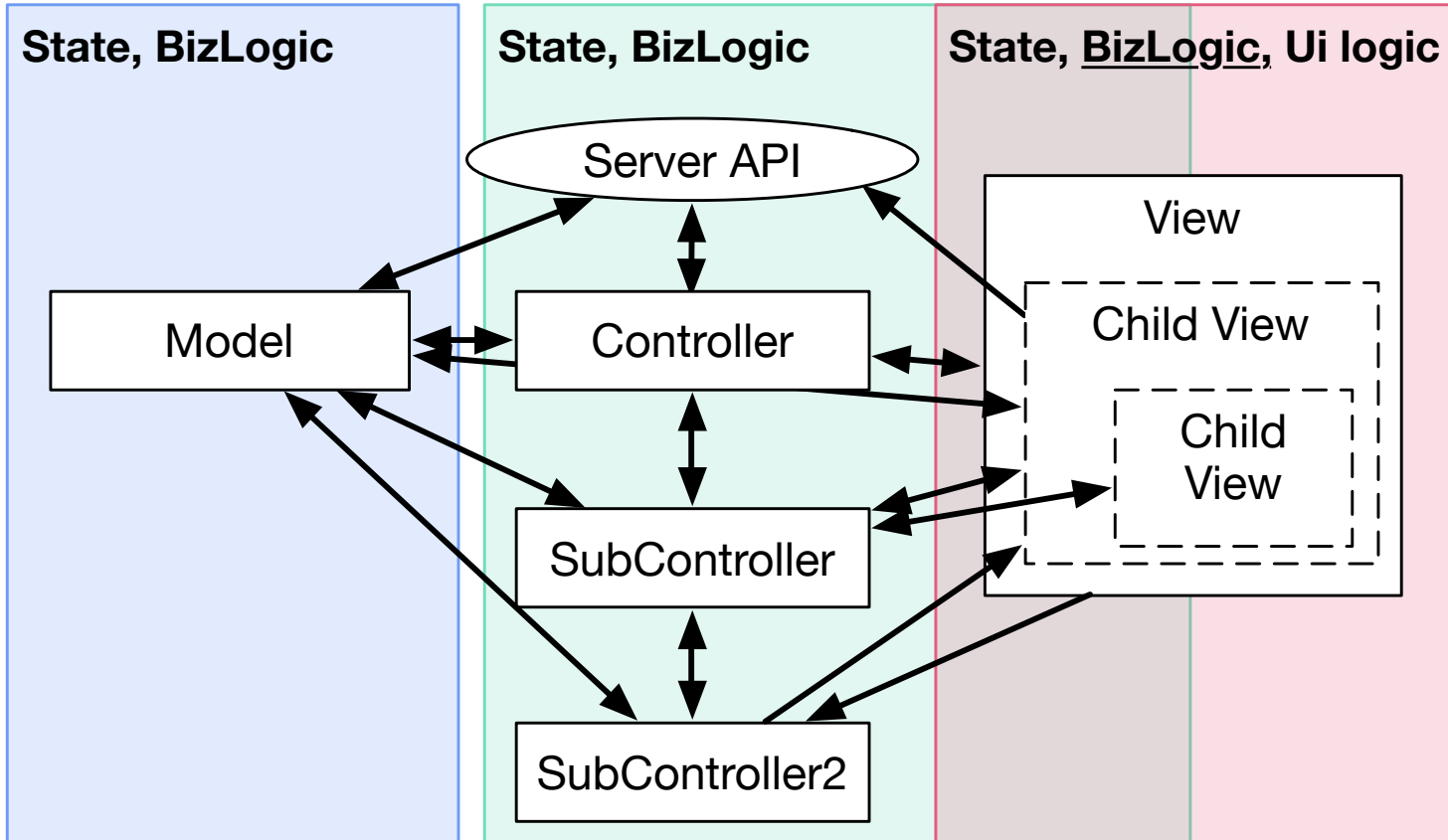
1. Store'ы

- Разные зоны ответственности
- Store -> область хранения данных (Store) и отдельно логика (Controller) (сообразно тому, [куда идет развитие Flux](#))

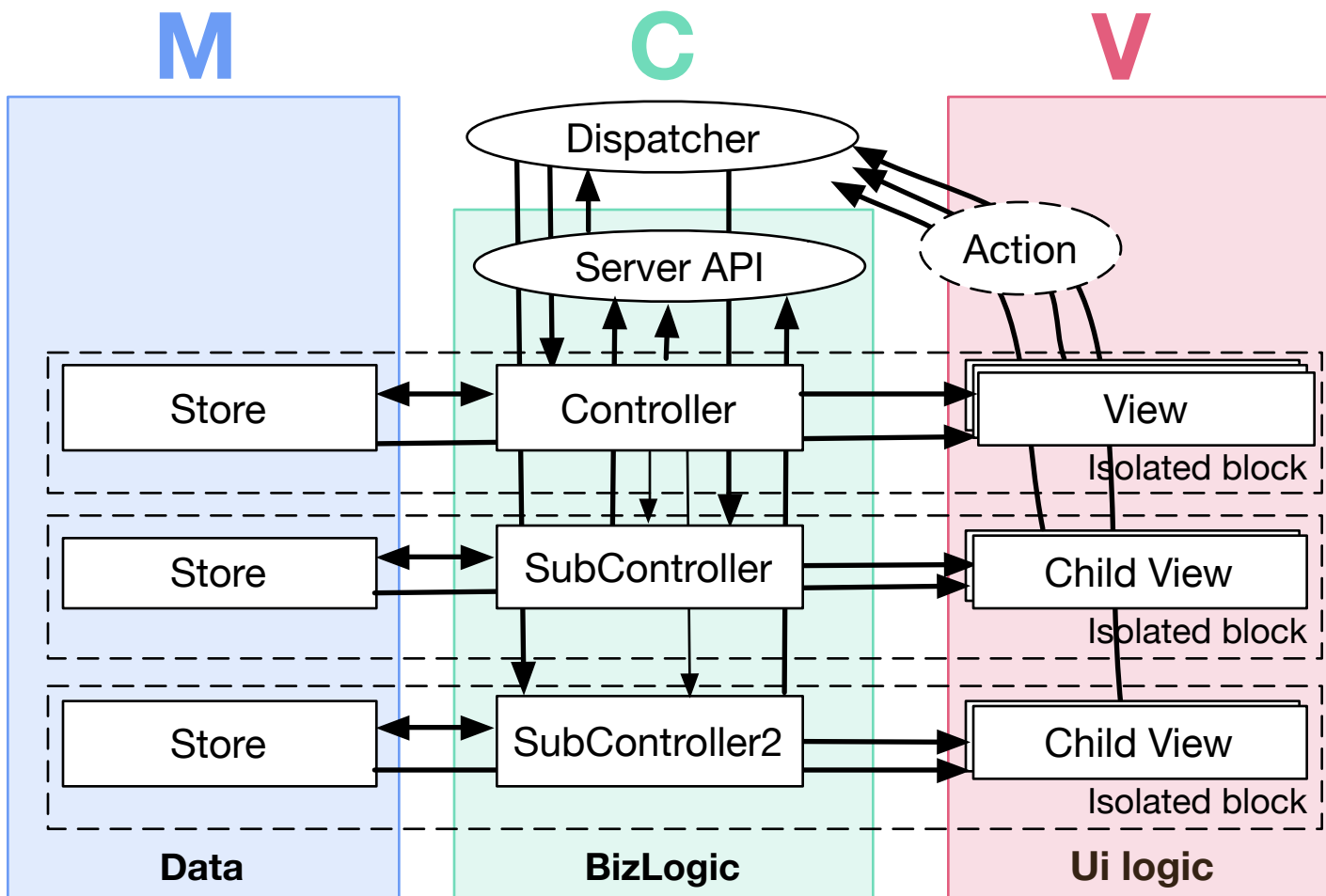
M

C

V



Примерно как было



Примерно как станет

2. JSX

- JSX — это опять мешанина кода и HTML.
- Обратили внимание на [wix-react-templates](https://wix-react-templates.com/)
- Написали свой похожий

```
return (  
  <div>  
    <h3>TODO</h3>  
    <TodoList items={this.state.items} />  
    <form onSubmit={this.handleSubmit}>  
      <input onChange={this.onChange} value={this.state.text} />  
      <button>{'Add #' + (this.state.items.length + 1)}</button>  
    </form>  
  </div>  
)
```

Шаблон

```
<a class="navigation-btn {class} {active}" onclick="{handleClick}">
  <span class="text">{text}</span>
  <#EXPANDER>
    <span class="icon" onclick="{handleExpanderClick}"></span>
  </#EXPANDER>
</a>
```

JS-код шаблона

```
return React.createElement(
  "a",
  {
    className: "navigation-btn " + (vars.class || "")
      + " " + (vars.active || ""),
    onClick: (vars.handleClick || "")},
  React.createElement(
    "span",
    {className: "text"},
    (vars.text || "")
  ),
  block(vars.EXPANDER, function (vars) {
    return (React.createElement(
      "span",
      {
        className: "icon",
        onClick: (vars.handleClick || "")}
    ));
  })
);
```

UI-КОМПОНЕНТ

```
type Props = {
  active: boolean,
  className: string,
  title: string,
  onClick: Function,
  onExpand: Function
};

type State = {
};

/**
 *
 */
export default class AppNavigationButton extends React.Component<Props, State> {

  render() {
    var vars = {
      active: this.props.active ? 'active' : '',
      'class': this.props.className,
      text: this.props.title,
      handleClick: this.props.onClick
    };

    if (this.props.onExpand) vars['EXPANDER'] = {
      handleExpanderClick: this.props.onExpand
    };

    return template(vars);
  }
}
```

3. Переводы (i18n)

- Все Flux-фреймворки обошли эту тему стороной

```
"This is a demo page for all our UI components"  
  = "This is a demo page for all our UI components"  
"Sortable, selectable, column toggling"  
  = "Sortable, selectable, column toggling"
```



```
"This is a demo page for all our UI components"  
  = "Демо-страница UI компонентов"  
"Sortable, selectable, column toggling"  
  = "Сортируются, выбираются, отключаются колонки<span class='a' />"
```


Шаблон

```
React.createElement(  
  "div",  
  null,  
  React.createElement(  
    "h2",  
    null,  
    "Customized Grid"  
  ),  
  React.createElement(  
    "h4",  
    null,  
    "Сортируются, выбираются, отключаются колонки",  
    React.createElement(  
      "span",  
      {className: "a"}  
    )  
  ),  
  React.createElement(  
    "div",  
    {style: {}},  
    React.createElement(  
      CustomGrid,  
      null  
    )  
  )  
),
```

4. Динамика

- Нет динамического создания Store'ов и View-компонент.
- Как организовать независимую работу двух одинаковых блоков на одной странице?

Что получилось?

Получилось:

- Хороший ООП с интерфейсами и дженериками
- С dependency injection
- Только две внешние библиотеки – React и lodash
- Можно поменять что угодно
- С нормальной сборкой

 Ext JS 6

vs.

 React



Битва «ПИЛОТОВ»

Битва «ПИЛОТОВ»

Обрезанная копия существующей админки «с нуля»:

- ExtJS 6 + TypeScript
- Flux + React + Typescript

Сложности:

- анимации
- кастомный скроллбар
- интерфейс меняется для узких экранов
- мобильная версия
- JSON-REST API с авторизацией

Цифры*

	ExtJS6 demo	Flux+React demo
PRODUCTION BUILD JS CODE SIZE	1,45 MB	336 KB
PRODUCTION BUILD CSS CODE SIZE	345 KB	19.9 kB
# OF HTML DOM NODES	841	301
LOAD TIME (PRODUCTION, NO CACHE)	1.54 s	0,59 s
LOAD TIME (PRODUCTION, CACHE)	1.42 s	0,58 s
TIME UNTIL FIRST API REQUEST	0,405 s	0,168 s
JS INIT TIME (GOOGLE CHROME)	0,345 s	0,158 s
PRODUCTION BUILD MEMORY USAGE (GOOGLE CHROME)	24.2 Mb	11.8 Mb

* В реальном проекте разница в объемах кода и скорости инициализации будет меньше

Переход



Оценка работы над UI

А что понадобится?

- 60 простых
- 15 сложных

<input type="checkbox"/>	Name	Backup account	Status	Next backup
<input type="checkbox"/>	60761 - vz1 - 0 - mms - 2960	Ivan Ivanov(su@test.com)	⊘	N
<input checked="" type="checkbox"/>	asn2k8	Ivan Ivanov (su@test.com)	✔	O
<input checked="" type="checkbox"/>	dsdsd	Ivan Ivanov (su@test.com)	✔	O

- ✔ Backup account
- ✔ Status
- Last backup
- ✔ Next backup

Search ✕



Search **Error** ✕

- Error
- Warnings
- No errors
- Online
- Offline

SOME TEXT

- Backup

- Recovery

- Overview

- Activities

- Delete

- Add

Acronis

RESOURCES

UPDATES

POWERED BY
ACRONIS ANYDATA ENGINE

ABOUT



Placeholder ▼

One

Two

Three

SOME TEXT

All machines



Оценка работы над UI

- 10% = 2000 строк кода HTML+CSS+TS
- Заняло 4 дня 1 человека

Технология миграции

Варианты:

1. Новые проекты пишем на новом фреймворке.
2. Старый код не трогаем, новый встраиваем «независимыми блоками» – как `iframe` или кастомный UI-компонент для ExtJS.
3. При модификации старого кода – или правь, как есть, или портируй.



Спасибо! Вопросы?

 @ryba_xek

 ryba.xek

 s@averin.ru

Слайды,
предыдущие доклады:

<http://averin.ru/slides/>

<http://slideshare.net/rybaxek>