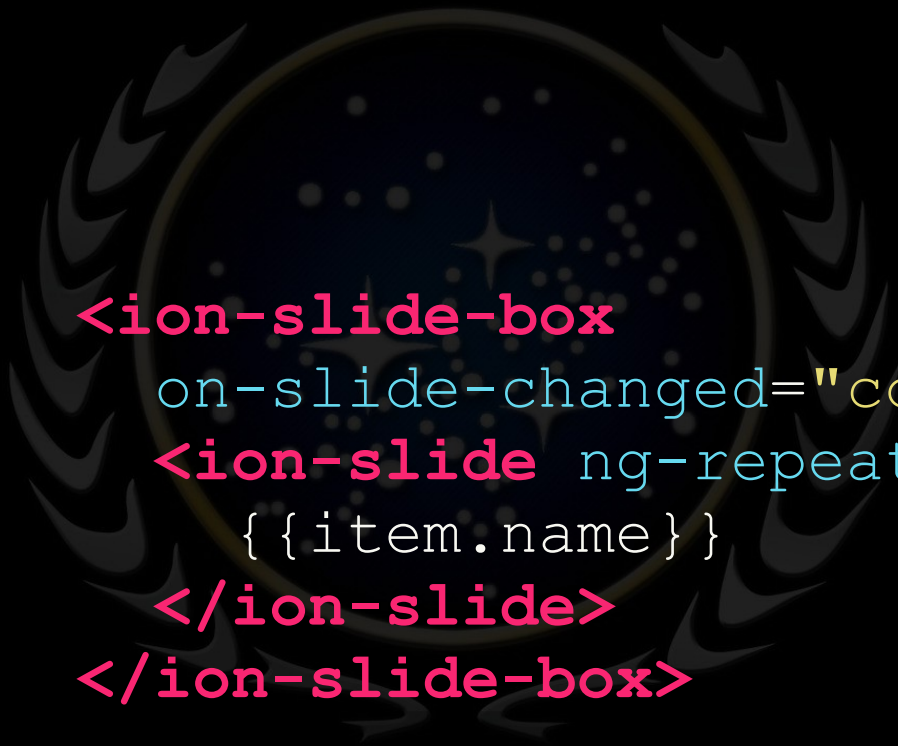
Developing Custom Directives

(when to do it and when not to)

Yuri Takhteyev, rangle.io
http://yto.io
@qaramazov

Directives I've Know and Loved



```
<ion-slide-box
  on-slide-changed="content.switchSlide(index)">
  <ion-slide ng-repeat="item in content.items">
    {{item.name}}
  </ion-slide>
</ion-slide-box>
```

But should you do it at home?

Directives vs what?

Skinny controllers & skinny directives

DO NOT

```
$scope.placeOrder = function() {
  var total = 0;
  angular.each($scope.items,function(item) {
    total = item.price * item.quantity * HST;
  });
  $http.post('/api/orders', {
    chargeAmoung: total
  })
    .success(function(...) {
      ...
    })
    .catch(function(...) {

    });
}
```

DO

```
$scope.placeOrder = function() {
  // Adjust the UI
  orders.placeOrder($scope.items)
    .then(function() {
      // Adjust the UI
    })
    .then(null, function(error) {
      // Adjust the UI
    });
}
```

☛ Same for directives!

Traditional Controllers

```
<div ng-controller="EditorCtrl">
  <div ng-controller="DeleteDialogCtrl">
    <ul ng-controller="ContentCtrl">
      <li ng-repeat="item in items">
        {{item.name}}
        <button ng-click="edit(item)">
          Edit
        </button>
        <button ng-click="showDeleteDialog(item)">
          Delete
        </button>
      </li>
    </ul>
  </div>
</div>
```

“Controller as...”

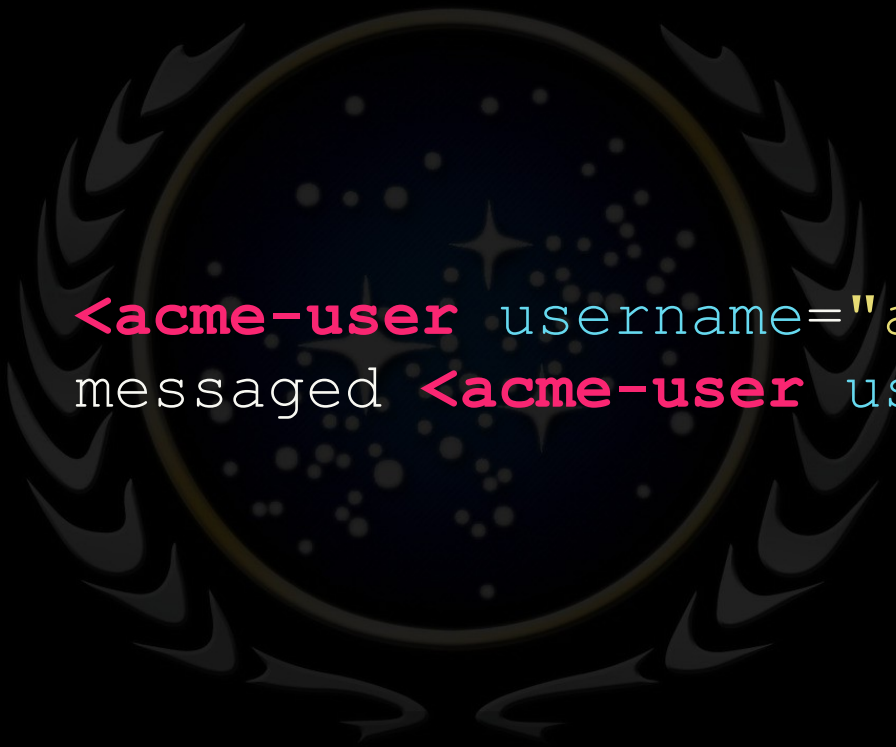
```
<div ng-controller="EditorCtrl as editor">
  <div ng-controller=
    "DeleteDialogCtrl as deleter">
    <ul ng-controller="ContentCtrl as content">
      <li ng-repeat="item in content.items">
        {{item.name}}
        <button ng-click="editor.show(item)">
          Edit
        </button>
        <button ng-click="deleter.show(item)">
          Delete
        </button>
      </li>
    </ul>
  </div>
```

"Good" controllers vs directives

- ☛ Multiple use
- ☛ DOM manipulation *
- ☛ Degree of isolation

* If you must...

Reuse



```
<acme-user username="alice"></acme-user>
messaged <acme-user username="bob"></acme-user>
```

- ☛ Consider ng-repeat
- ☛ Consider ng-include
- ☛ Consider ui-router

Integrated

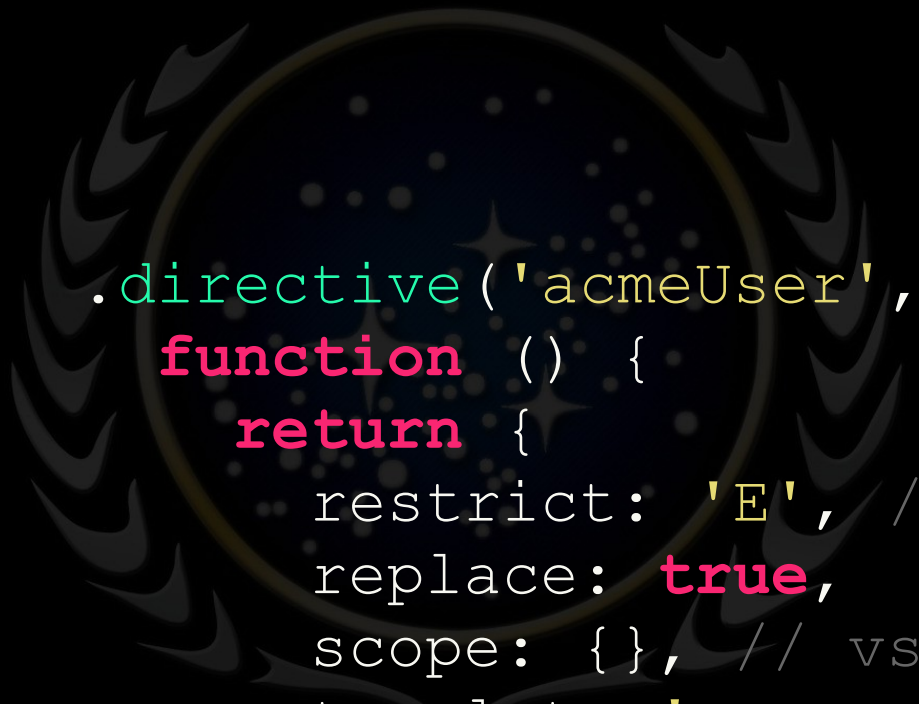
```
<div ng-controller="EditorCtrl">
  <div ng-controller="DeleteDialogCtrl">
    <ul ng-controller="ContentCtrl">
      <li ng-repeat="item in items">
        {{item.name}}
        <button ng-click="edit(item)">
          Edit
        </button>
        <button ng-click="showDeleteDialog(item)">
          Delete
        </button>
      </li>
    </ul>
  </div>
</div>
```

Isolated

```
<div ng-controller="EditorCtrl">
  <div ng-controller="DeleteDialogCtrl">
    <ul ng-controller="ContentCtrl">
      <li
        ng-repeat="item in items"
        acme-item="{{item}}"></li>
    </ul>
  </div>
</div>
```

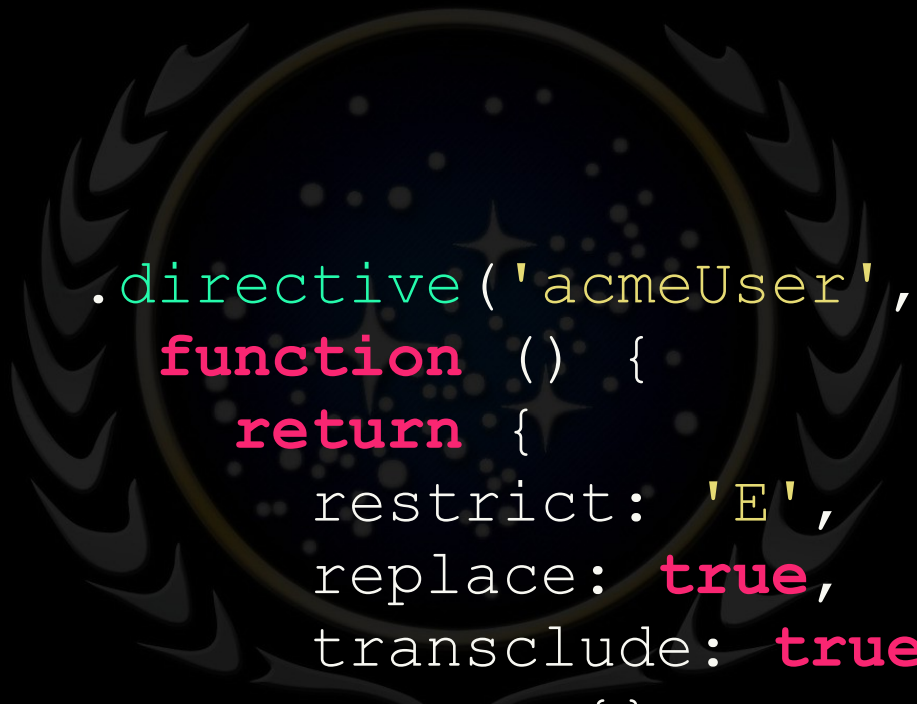
☛ But is it?

Defining a Really Basic Directive



```
.directive('acmeUser',
  function () {
    return {
      restrict: 'E', // vs 'A', 'AE'
      replace: true,
      scope: {}, // vs 'true', 'null'
      template:'<span>user</span>'
    };
  }
)
```

Transclusion



```
.directive('acmeUser',
  function () {
    return {
      restrict: 'E',
      replace: true,
      transclude: true,
      scope: {},
      template:'<span>user ' +
        '<div ng-transclude/></span>'
    };
  }
)
```

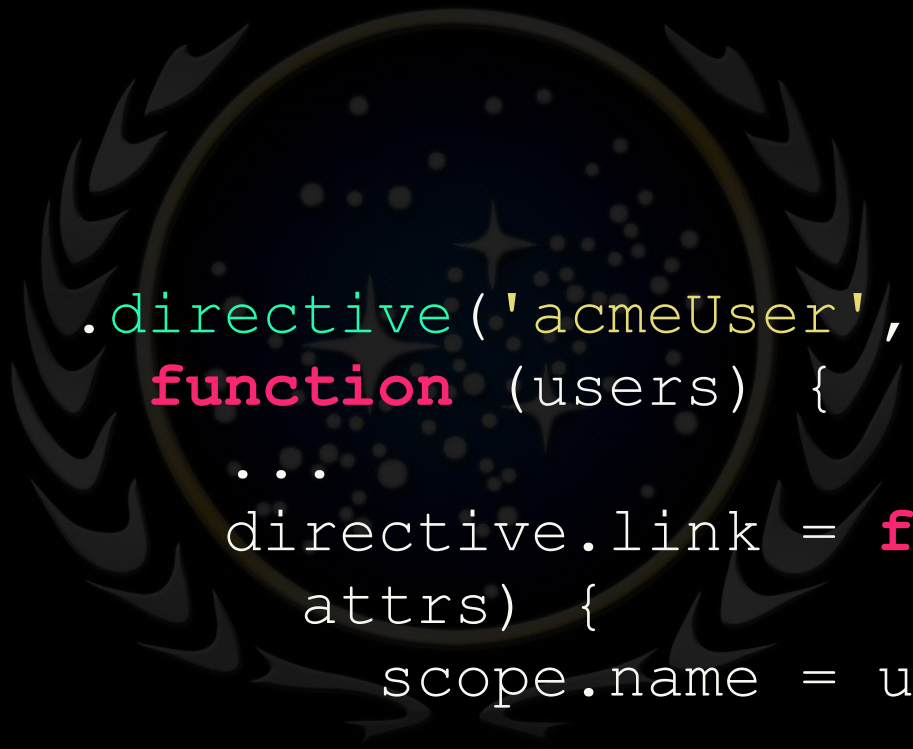
An External Template

```
.directive('acmeUser',
  function () {
    return {
      restrict: 'E', // vs 'A', 'AE'
      replace: true,
      scope: {},
      templateUrl:'/users/user.html'
    };
  }
)
```

Linking

```
.directive('acmeUser',
  function () {
    var directive = {
      restrict: 'E',
      replace: true,
      scope: {},
      templateUrl: '/user/user.html'
    };
    directive.link = function(scope, element,
        attrs) {
      ...
    };
    return directive;
  }
)
```

External communication: services



```
.directive('acmeUser', ['users',
  function (users) {
    ...
    directive.link = function(scope, element,
      attrs) {
        scope.name = users.getName();
        ...
    };
    ...
  }
])
```

External communication: attributes, string

```
<acme-user username="alice"></acme-user>
```

```
directive.scope = {
  username: '@username'
};

directive.link = function(scope, element, attrs){
  scope.user = users.getUser(scope.username);
};
```

External communication: attributes, 2-way-binding

```
<acme-user username="user"></acme-user>
```

```
directive.scope = {
  username: '=username'
};

directive.link = function(scope, element, attrs){
  scope.user = users.getUser(scope.username);
};
```

External communication: binding callbacks

```
<acme-user username="user"
  on-ban="handleBan(user)"></acme-user>

directive.scope = {
  fireBan: '&onBan'
};

directive.link = function(scope, element, attrs){
  ...
  scope.fireBan();
  ...
};
```

☛ Consider using a service

External communication: attribute processing

```
<acme-user username="{{user}}"></acme-user>


directive.scope = {};

directive.link = function(scope, element, attrs){
  ...  attrs.username ...
};
```

External communication: $parsing expressions

```
<acme-user username="{{user}}"
  cost="hours * rate"></acme-user>

directive.link = function(scope, element, attrs) {
  var userData = users.getUser(scope.username);
  var getCost = $parse(attrs.cost);
  scope.cost = getCost({
    rate: userData.rate,
    hours: userData.cost,
    discount: userData.discount
  });
};
```

When to use compile: rarely

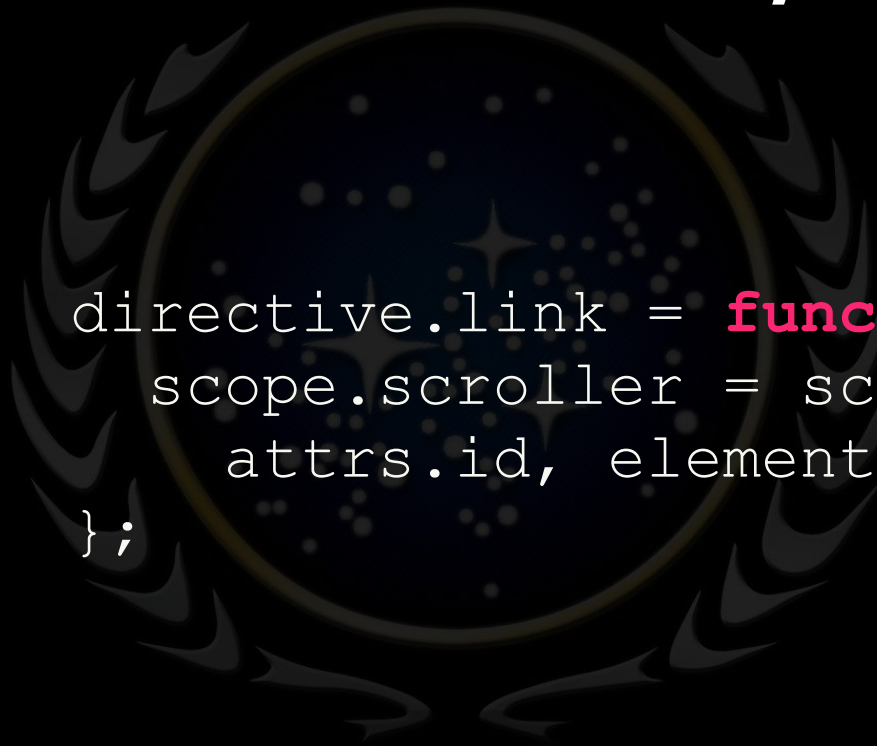
```
<acme-user username="{{user}}"
  cost="hours * rate"
  repeat="5"></acme-user>

directive.compile = function (tElement, tAttrs) {
  var wrapper = angular.element('<div></div>');
  for (var i=0; i<tAttrs.repeat; i++) {
    wrapper.append(tElement.clone());
  }
  tElement.replaceWith(wrapper);
  return function (scope, iElement, iAttrs) {
    ...
  };
};
```

Directives & Services

```
$ionicSlideBoxDelegate.$getByHandle('users')
  .update();
```

☛ Use a service to control a directive

Directives, Services & DOM



```
directive.link = function(scope, element, attrs) {
  scope.scroller = scroller.makeScroller(
    attrs.id, element);
};
```

☛ Offload DOM processing to a service

Testing Directives

```
element = $compile(template)(scope);
element.click();
user.select.should.have.been.calledOnce;
```

☛ But focus on services!

Thank You.

Contact:

yuri@rangle.io

http://yto.io

@qaramazov

This presentation:

http://yto.io/xdir

Image Credits

by jscolemanfch