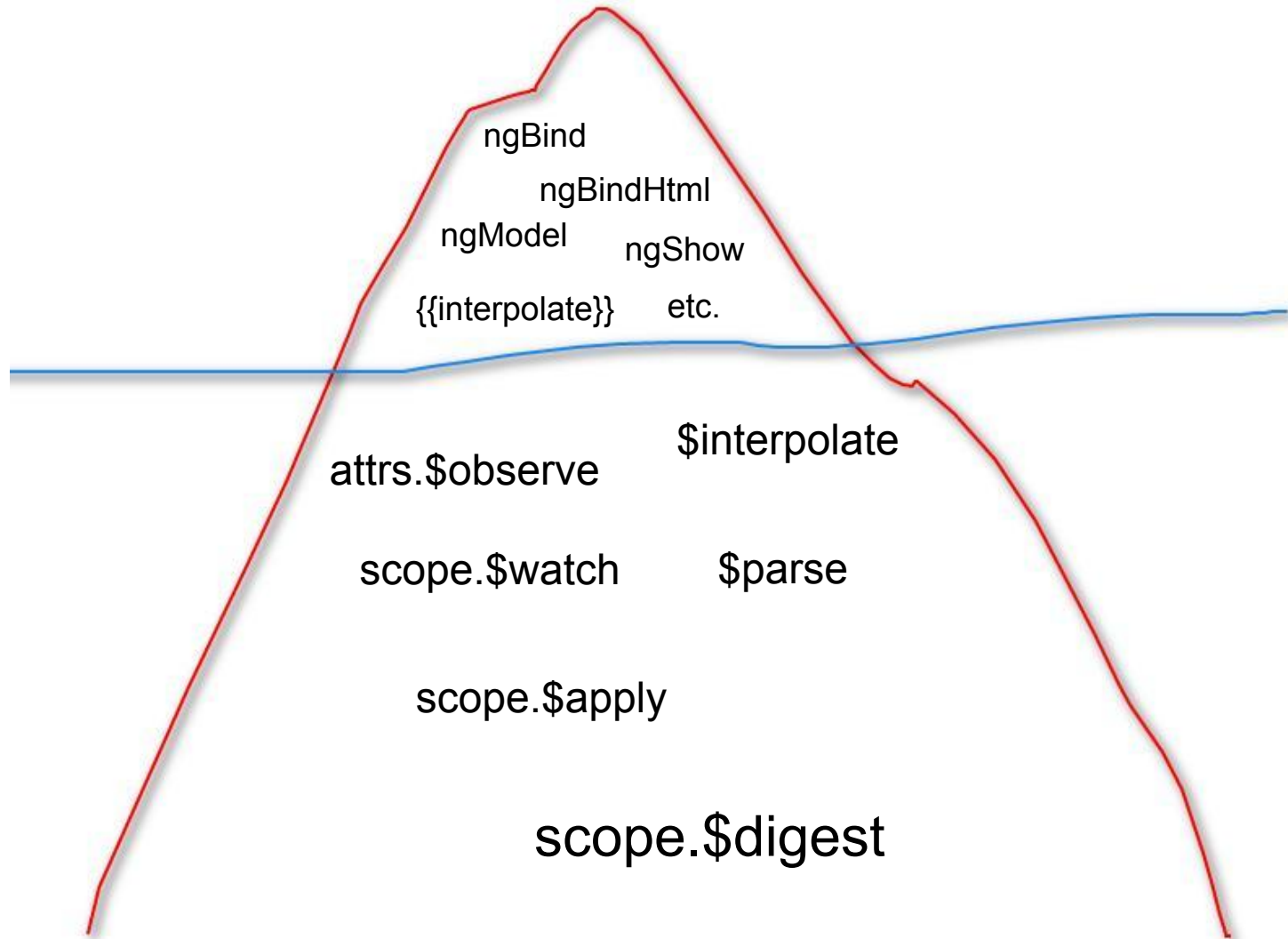


\$digest

@buunguyen



Search

\$apply already in progress

search

459 results

10 Q: \$apply already in progress

Search

\$digest already in progress

146 results

relevance

newest

Search

\$digest() iterations reached

search

e.\$a
the c
the s

132 results

0 votes

1 answer

Q: 10 I'm receiving an error: Wh
angular



stackoverflow

Questions

Tags

Users

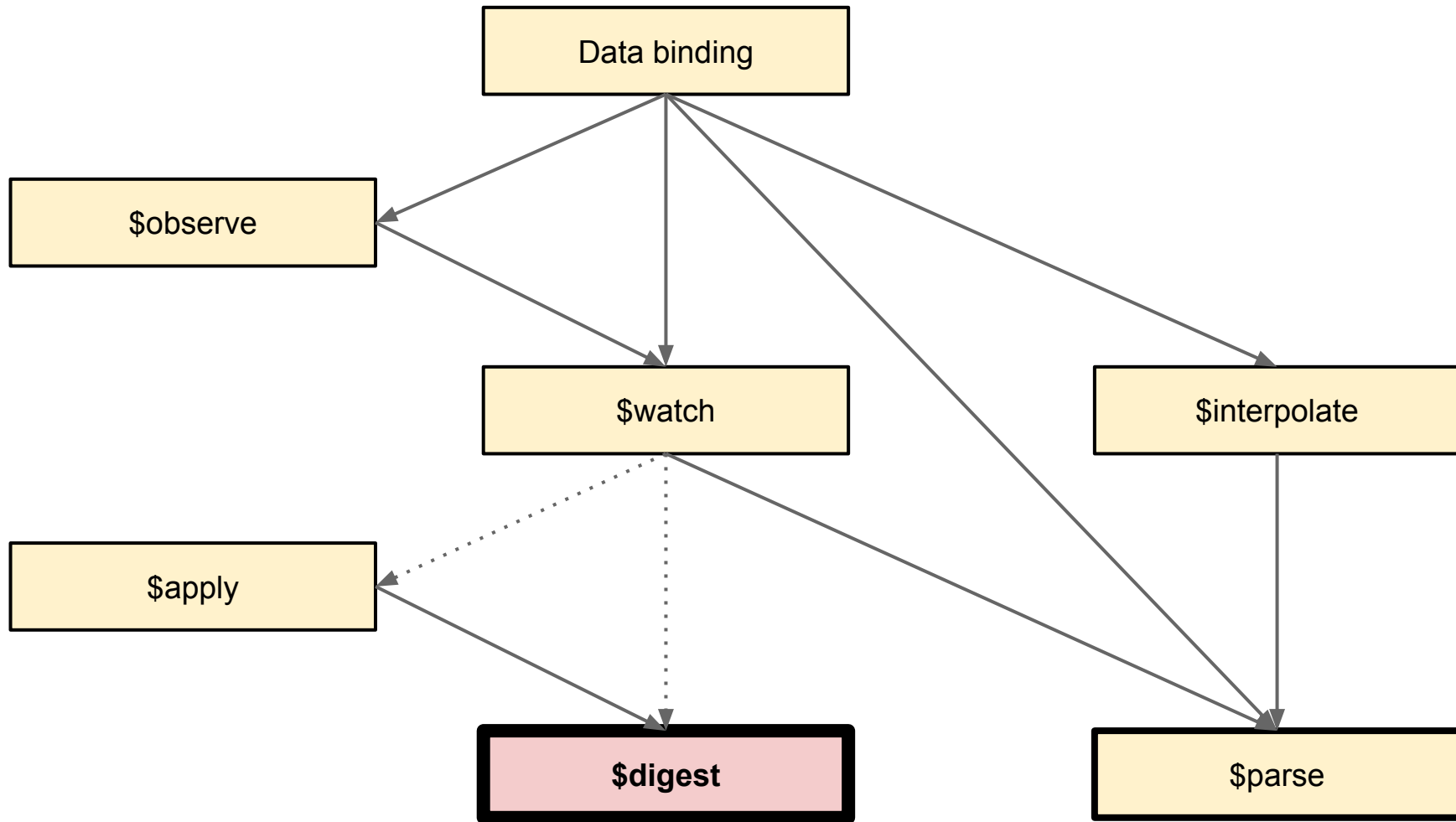
Badges

U

Controller function getting called twice using ng-show

1

Q: Angularjs 10 \$digest() iterations reached



- **Binding by \$watch, \$observe, \$parse, \$interpolate**
 - **\$interpolate by \$parse**
 - **\$observe by \$watch**
 - **\$watch by \$parse, \$apply, \$digest**
 - **\$apply by \$digest**

Demo: a_bindings.html

Demo: b_bindings.html

Relevant AngularJS source: ngBind.js, input.js, interpolate.js, parse.js, compile.js

- **Binding by \$watch, \$observe, \$parse, \$interpolate**
 - **\$interpolate by \$parse**
 - **\$observe by \$watch**
 - **\$watch by \$parse, \$apply, \$digest**
 - **\$apply by \$digest**

Relevant AngularJS source: rootScope.js

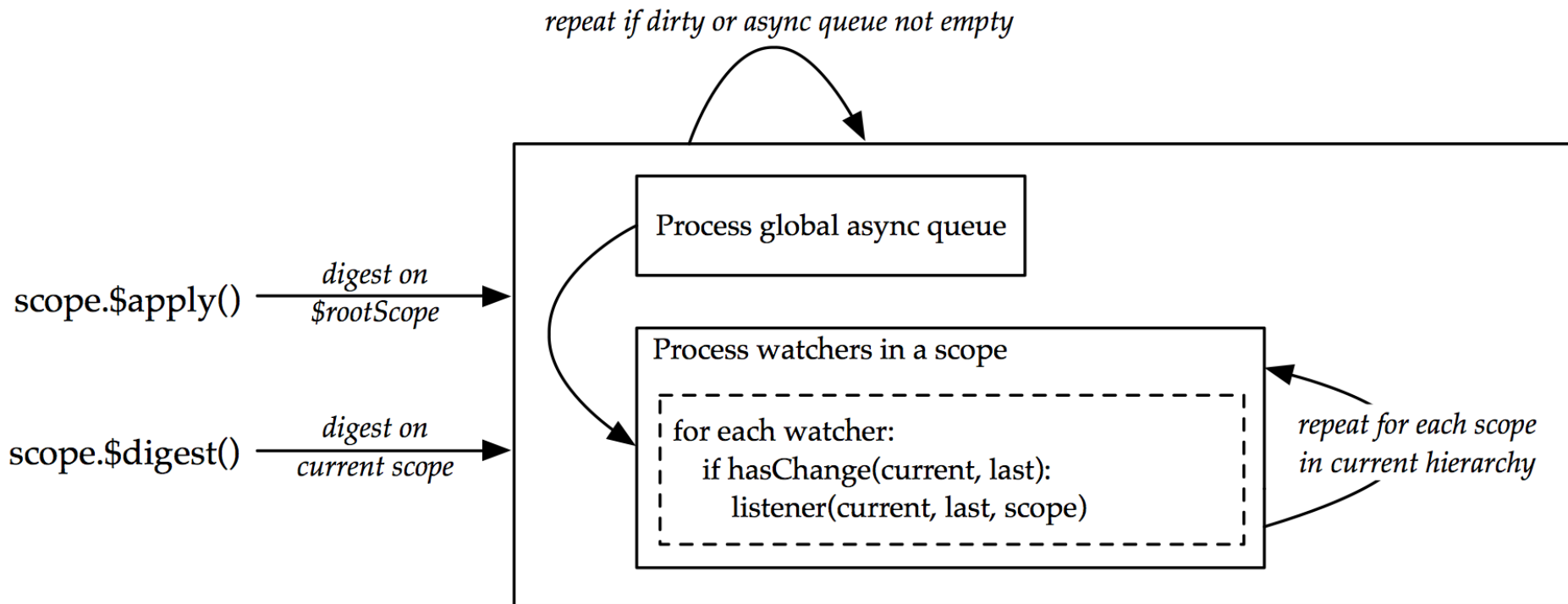
<pre> <!DOCTYPE html> ▼<html ng-app class="ng-scope"> ►<head>...</head> ▼<body> ▼<div> <div ng-controller="GreetCtrl" class="ng-scope ng-binding"> Hello World! </div> ▼<div ng-controller="ListCtrl" class="ng-scope"> ▼ <!-- ngRepeat: name in names --> <li ng-repeat="name in names" class="ng-scope ng-binding"> Igor <li ng-repeat="name in names" class="ng-scope ng-binding"> Misko <li ng-repeat="name in names" class="ng-scope ng-binding"> Vojta </div> </div> </body> </html> </pre>	<pre> \$scope \$scope name='Wold' \$scope names=[...] \$scope name='Igor' \$scope name='Misko' \$scope name='Vojta' </pre>
http://docs.angularjs.org/guide/scope	

```
function Scope() {  
  this.$$watchers = []  
}
```

```
Scope.prototype.$watch = function(exp, listener, deep) {  
  this.$$watchers.push({  
    getter : $parse(exp),  
    last   : undefined,  
    fn     : listener,  
    deep   : deep  
  })  
}
```

(vastly simplified)

Relevant AngularJS source: rootScope.js



```
function beginPhase(phase) {
  if ($rootScope.$$phase)
    throw new Error($rootScope.$$phase + ' already in progress')
  $rootScope.$$phase = phase
}
```

```
Scope.prototype.$apply = function() {
  beginPhase('$apply')
  try {
    return this.$eval(expr) // => $parse(expr)(this)
  } finally {
    clearPhase()
    $rootScope.$digest() // run on $rootScope
  }
}
```

```
Scope.prototype.$digest = function() {
  beginPhase('$digest')
  try {
    // DIGEST LOOP FOR CURRENT SCOPE
  } finally {
    clearPhase()
  }
}
```

Demo: [c_apply_safeapply.html](#)

Demo: [d_digest_vs_apply.html](#)

Relevant AngularJS source: [rootScope.js](#)

**\$digest keeps iterating until models
stabilize**

Demo: [e_model_stabilizes.html](#)

Relevant AngularJS source: [rootScope.js](#)

What we've learned

- 2-way binding under the hood
- Inner working of digest loop
- Performance pitfalls and bugs

Thanks! Find me @buunguyen