Authentication + Authorization Architecture

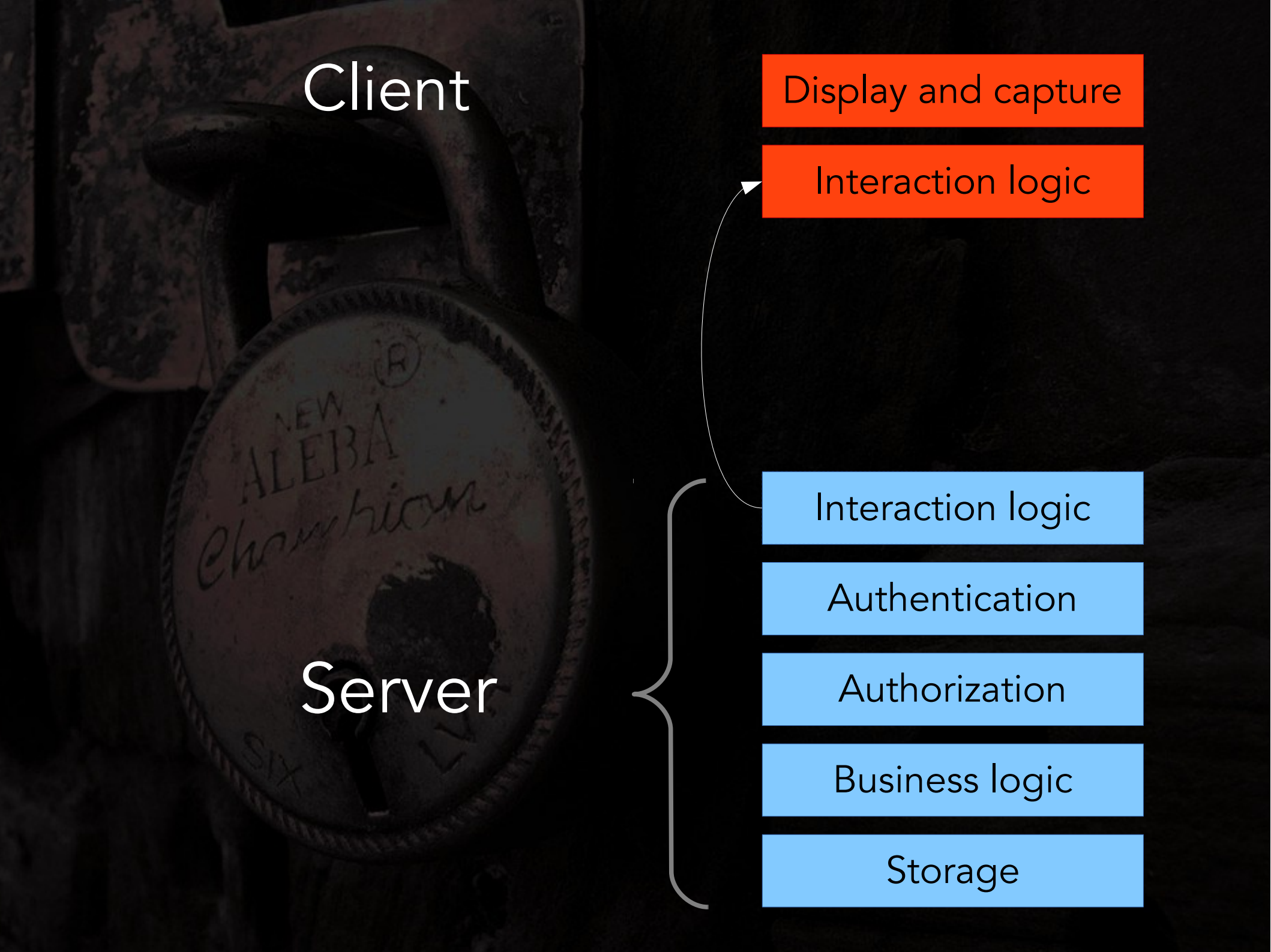
in Browser Applications

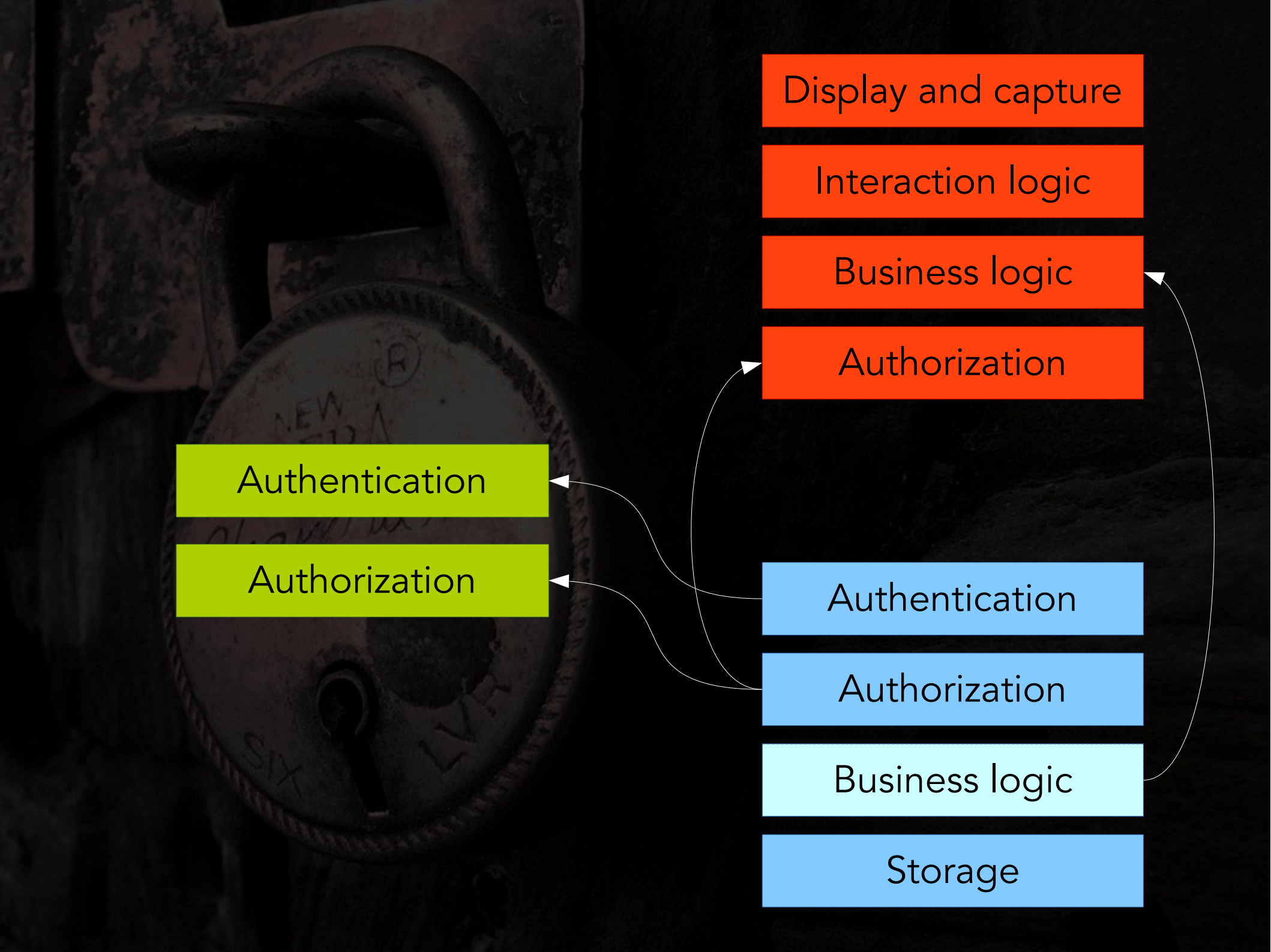
Yuri Takhteyev, rangle.io
http://yto.io
@qaramazov

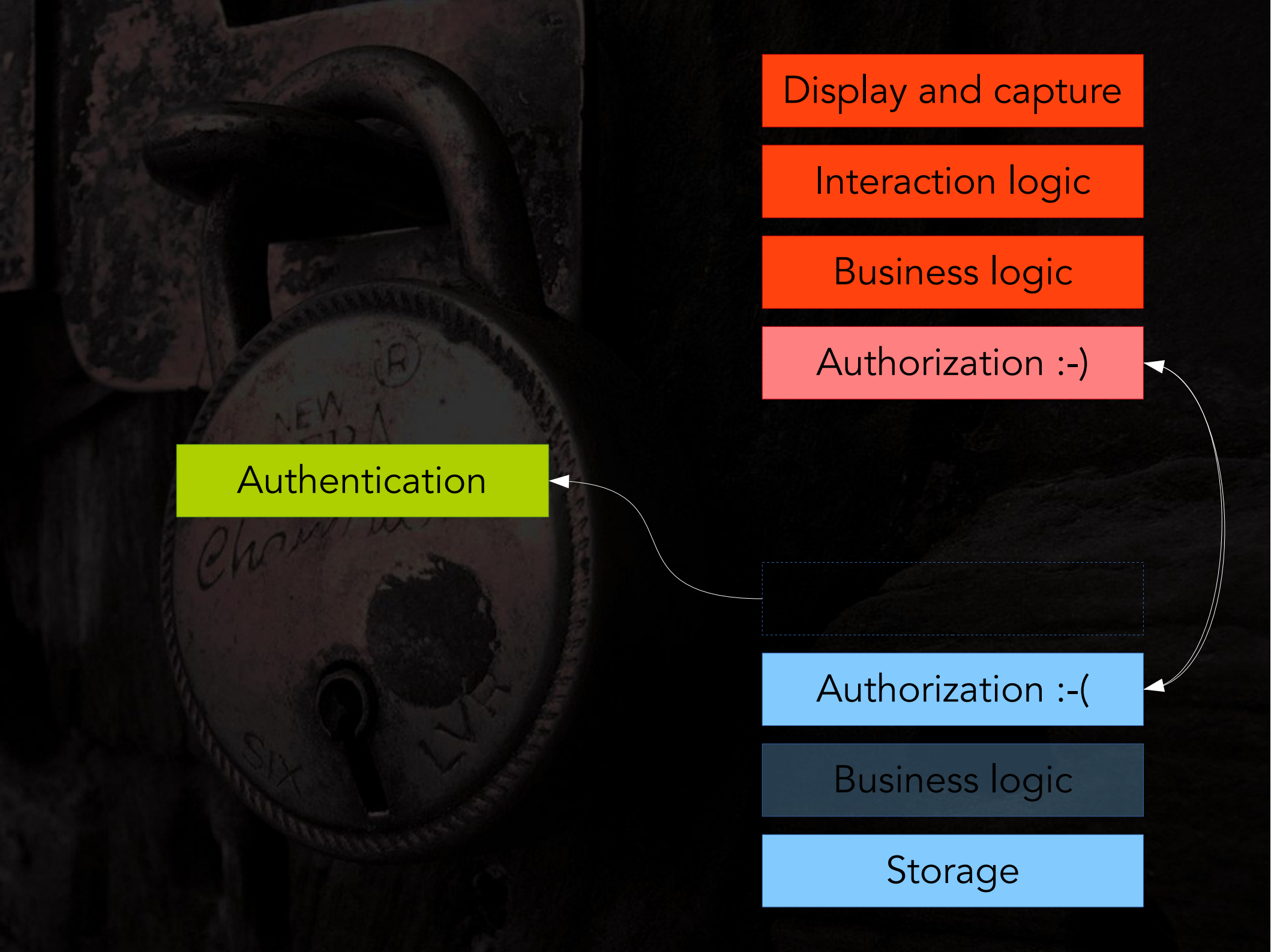
Authenticate, Authorize

Protecting clients from:

- outsiders
- each other
- themselves


Client
Display and capture
Interaction logic
Server
Interaction logic
Authentication
Authorization
Business logic
Storage


Display and capture
Interaction logic
Business logic
Authorization
Authentication
Authorization
Authentication
Authorization
Business logic
Storage


Display and capture
Interaction logic
Business logic
Authorization :-)
Authentication
Authorization :-(
Business logic
Storage

Who am I?

Cookies
vs tokens

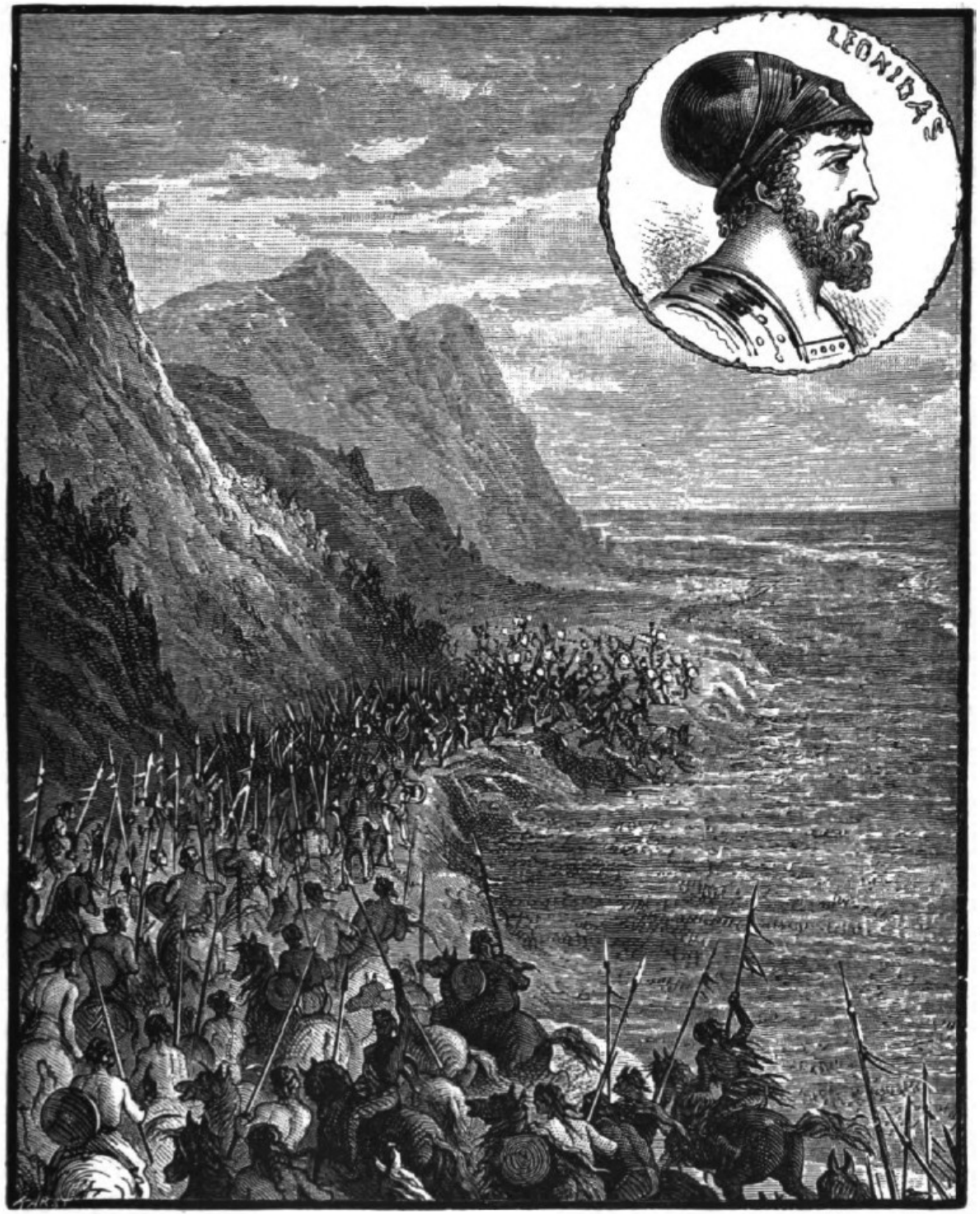
Maintaining
the state

Passport

- Local
- Social / OAuth

Many Faces of Authorization

- By role or activity
- By resource instance
- Altogether custom



Bottlenecks

PUBLIC

rangle / koast

A thin server with an attitude — Edit

29 commits | 3 branches | 3 releases | 1 contributor

branch: master | koast /

Updating READMEs.

yuri authored a minute ago — latest commit 9baedeadcc

client	Updating READMEs.	a minute ago
docs	Updating heroku docs.	a month ago
examples	Updating READMEs.	a minute ago
server	Updating READMEs.	a minute ago
.bowerrc	Adding support for persona authentication.	24 days ago
.gitignore	Switching to OAuth authentication via passport.	16 days ago
Gruntfile.js	JSHinting and beautifying	2 months ago
LICENSE.txt	Updating the READMEs.	a month ago
README.md	Updating READMEs.	a minute ago
bower.json	Switching to OAuth authentication via passport.	16 days ago

README.md

Koast provides a base for a backend server to support an AngularJS app. (It may be useful for Javascript apps built with other frameworks as well, but our focus is on Angular.) The goal is to quickly

SSH clone URL

git@github.com:rang

You can clone with HTTPS, SSH, or Subversion.

Setting Up an API Route

```
var mapper = koast.makeMongoMapper(connection);

routes = [
  ['get', 'robots', user.any,
    mapper.get('robots', [])
  ],
  ['put', 'robots/:robotNumber', user.any,
    mapper.put('robots', ['robotNumber'])
  ]
];
```

Restricting Access by Object

```
mapper.queryDecorator = function(query, req,
  res) {
  query.owner = req.user.username;
};
```

```
mapper.queryDecorator = function(query, req,
  res) {
  query.clientId = req.user.clientId;
};
```

Post-Query Filtering

```
mapper.filter = function(result, req) {
  if (req.method === 'GET' {
    return canSee(data, req);
  } else {
    return canEdit(data, req);
  }
};
```

Informing the Client

```
mapper.clientAuthorizer = function(result, req) {
  result.meta.can.edit =
    canEdit(result.data, req);
};
```

The Client Side

```
angular.module('sampleKoastClientApp',
['koast'])

.controller('myCtrl', ['$scope', 'koast',
'$log',
  function($scope, koast, $log) {

    $scope.login = function(provider) {
      koast.user.initiateOauthAuthentication(
        provider);
    };

    ...
```

The Client Side

```
...

koast.user.getStatusPromise()
  .then(function() {
    if (koast.user.isAuthenticated) {
      return koast.getResource('robots')
        .then(function(robots) {
          $scope.robots = robots;
        });
    }
  })
  .then(null, $log.error);
}]);
```

The Template

```
<span ng-if="robot.can.edit">
 <button>...</button>
</span>
```

Thank You.

Contact:

yuri@rangle.io

http://yto.io

@qaramazov

This presentation:

http://yto.io/auth

Koast:

http://yto.io/koast

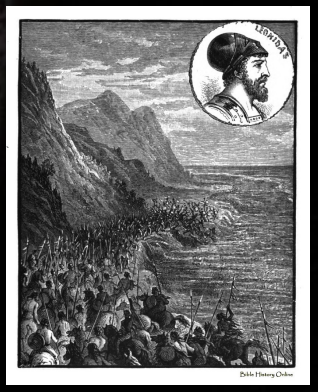
Image Credits

by psd

by nikhilverma

by dunechaser

by yewenyi



by bible-history

by striatic