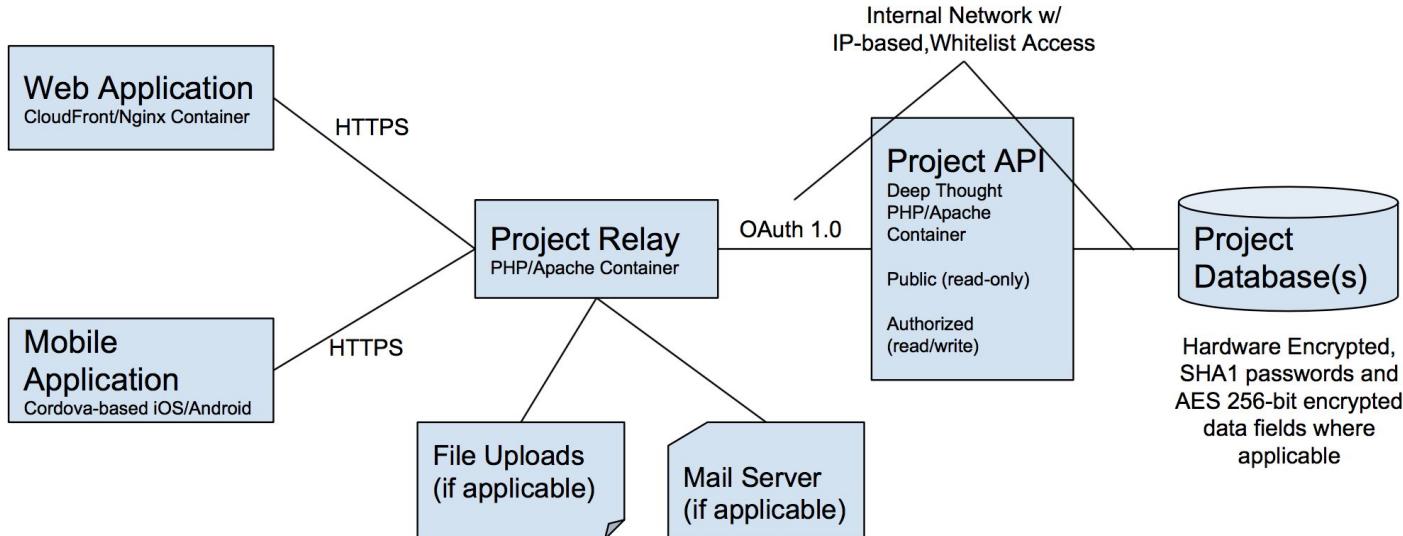


# Confide

Single-Page Application w/ Secure API Communication Using DTRelay

# DTRelay Sample Architecture



Amazon Web Services EC2s + RDS (Postgres/MySQL) in Docker Swarm Configuration

# API Configuration

Connect to Public or Secure/OAUTH-based APIs

```
1  {
2    "api": {
3      "url": "http://api.local/api/",
4      "delegates": [
5        "DTProviderDelegate",
6        "MFADelegate"
7      ],
8      "consumer_key": "██████████",
9      "consumer_secret": "██████████"
10    },
11    "api-oauth": {
12      "url": "http://api.local/api-oauth/",
13      "delegates": [
14        "DTProviderDelegate",
15        "DTOAuth1Delegate",
16        "MediaDelegate"
17      ],
18      "consumer_key": "██████████",
19      "consumer_secret": "██████████",
20      "request_token_url": "http://api.local/api-oauth/user/request_token",
21      "authorize_url": "login",
22      "callback_url": "http://localhost/api-oauth",
23      "access_token_url": "http://api.local/api-oauth/user/access_token"
24    },
25    "aws-s3": {
26      "url": "us-west-2",
27      "consumer_key": "██████████",
28      "consumer_secret": "██████████",
29      "bucket": "dt-development",
30      "path": "sandbox"
31    },
32    "facebook": {
33      "url": "https://graph.facebook.com/v2.7/",
34      "delegates": ["DTOAuth2Delegate"],
35      "consumer_key": "██████████",
36      "consumer_secret": "██████████",
37      "authorize_url": "https://www.facebook.com/dialog/oauth",
38      "redirect_url": "http://localhost:8887/facebook/",
39      "access_token_url": "https://graph.facebook.com/oauth/access_token",
40      "response_type": "code",
41      "scope": "public_profile"
42    },
43    "twitter": {
44      "url": "https://api.twitter.com/1.1/",
45      "delegates": ["DTOAuth1Delegate"],
46      "consumer_key": "██████████",
47      "consumer_secret": "██████████",
48      "request_token_url": "https://api.twitter.com/oauth/request_token",
49      "authorize_url": "https://api.twitter.com/oauth/authorize",
50      "callback_url": "http://localhost:8887/twitter/",
51      "access_token_url": "https://api.twitter.com/oauth/access_token",
52      "handle": "@photrist"
53    },
54    "google": {
55      "url": "https://www.googleapis.com/oauth2/v2/",
56      "delegates": ["DTOAuth2Delegate"]}
```

# Multi-Factor Authentication

MFADelegate

## Please Confirm

You must confirm your access before you can log in. Please check your email for a Magic Link.

As a demo, your email will arrive at [MailHog](#).

[Close](#)

Password

....

[Login](#)

You will receive an email with a magic link to complete login. Once you have clicked the magic link, please confirm here.

[Confirm Magic Link](#)

Confide

Powered by DTRelay. Protection in Motion™  
Copyright © 2017 Expressive Analytics, LLC

this application is for demonstration purposes only and should not be used for medical images

# MFA Delegate

- Simple relay delegate to trigger multi-factor authentication (MFA)
- During standard authentication (user login), it handles MFA:
  - We have configured the delegate to require 2 factors of authentication
  - Step 1: send an *authorize\_factor* request for the user's ID
  - Step 2: send a magic link via user's email address
- We can easily support any authorizer that contains a valid identifier

```
40 public function postprocess(&$rsp,&$obj){  
41     global $config;  
42     if($this->cmr->endpoint=="authentication" && $this->cmr->action=="authenticate"){  
43         $tok = $obj["obj"];  
44         if($tok["user_id"]){ // we have successfully logged in  
45             $tok_str = $tok["token"];  
46  
47             // STEP 1: authorize "id" factor  
48             // Since there is a race condition for request_tokens, we need to save  
49             // the correct MFA token for future authorizers.  
50             $tgt = DTAPI::consumer("api-oauth");  
51             $this->cmr->keystore->set($tgt->api,"mfa_token",$tok_str);  
52             $this->cmr->action("authorize_factor")->post(array(  
53                 "key"=>"id","val"=>$tok["user_id"],"oauth_token"=>$tok_str  
54             ));  
55  
56             // STEP 2: send magic link via email  
57             $mailer = new DTPHPMailer(null,null);  
58             $to = new DTEmailRecipientList();  
59             $to->setAddresses(array("email"=>$this->email));  
60             $subject = "Please Confirm Login Attempt";  
61             $magic_link = $config["relay_url"].".api/authentication?act=authorize_factor&oauth_token={$tok_str}&key=magic_link";  
62             ob_start(); //capture the following as HTML  
63             require_once(dirname(__FILE__)."../email/magic_link.php");  
64             $html = ob_get_contents();  
65             ob_end_clean();  
66             $mailer->sendHTMLEmail($to,$subject,$html);  
67         }  
68     }  
69 }
```

# Secure Image Encryption

DTRelay's Secret +  
MediaDelegate

EyMWZhMjBhZjg0ZmZ1YjA1ZTkzY  
zY4NTZjYWNhNGUwNjQ1ZjdjZGJm  
OGewMWI4NjkwYzBhNWM2M2ExYjB  
hOGJmMjJmMjIxNWRiYTcyNzcxZT  
A3ZTUyOTY2MDN1NDg1MTQyODE5Y  
Tc2YTAwYWMxNGNhMjgwM2I0ZGMz  
ODg5MTIyNjM0YzQ3MGR1ZjRhY2V  
iMGNiMTIwMTUyZjkzODc2ZjBmZW  
ZkODRhMjdlZTBhOGE5Zj1jN2JiO  
WQ2ZmQxZDk0MTA2YjAxYzRmZDBm  
MjQ0M2RjNzY3YWZiM2JiMTIwNWY  
zYWJiYzczNjFmNjEzYTg4NTQzNz  
Y0MWE2OWQ0ZTkyNzljY2VkNmM4Y  
zNiOWQ4ZDg0a741DN1NDg1MTQyO  
DE5YTc2YTAwYWMxNGNhMjgwM2I0

# Protection in Motion™

- During initialization, the relay provides the client application with a secret
- We can use this secret value to encrypt our images during transfer
- The relay decrypts the image and processes it as a standard upload
- This same technique is used to encrypt any “protected parameters” so that they are tamper-proof and undecipherable in-transit (even without SSL)

# Client-side Encryption (AES)

Using DTRelay's secret value as the encryption key, and the request's unique nonce value as the initialization vector, we use AES to encrypt the images during transfer.

```
123   |     Relay.get("SecureMedia",params,function(response){  
124   |       $ctrl.item = response.obj // populate the object  
125   |     Relay.getAPIs().then(function(apis){  
126   |       var api = apis["SecureMedia"]  
127   |       var params = {  
128   |         "act":"download",  
129   |         "id":$ctrl.item.id,  
130   |         "api":api.api,  
131   |         "end":api.end  
132   |       }  
133   |       if(typeof($stateParams.review)!=="undefined")  
134   |         params.review = true  
135   |       var params = Relay.insertTokens(params)  
136   |       $http.get(api["url"],{"params":params})  
137   |         .then(function(response,headers){  
138   |           var sec = localStorage.getItem("RelaySecret")  
139   |           var data = CryptoJS.AES.decrypt({  
140   |             ciphertext:CryptoJS.enc.Hex.parse(response.data),  
141   |             salt:""  
142   |           },CryptoJS.enc.Utf8.parse(sec),{  
143   |             iv:CryptoJS.enc.Utf8.parse(params.dtnnc)  
144   |           })  
145   |           if(data.length>0)  
146   |             $ctrl.item.data64 = "data:image/"+$ctrl.item.file_ext+";base64,"  
147   |               "+data.toString(CryptoJS.enc.Utf8)  
148   |           })  
149   |         })
```