CIRCLE

MOBILE FIRST APPSEC

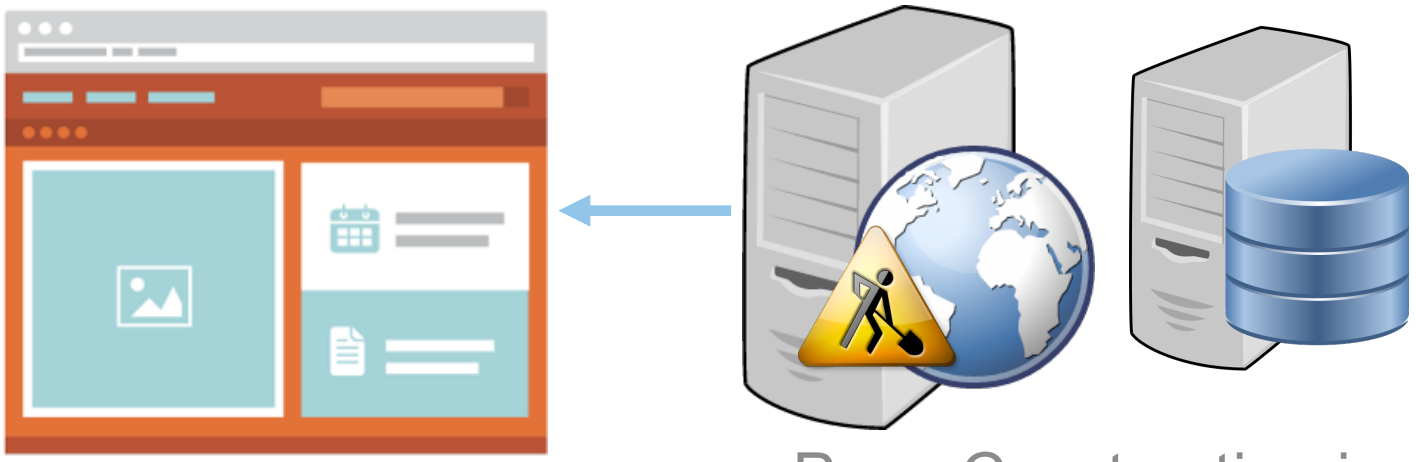
Scott Matsumoto, CISO

Boston Security Meetup

MOBILE FIRST APPSEC

- Mobile First Design
  - Design starts with Mobile
  - Transition from mobile to desktop
- Mobile First AppSec
  - Mobile First Apps Aren't Web-Apps
  - Different Page Construction
  - Different Technology Stack
  - Different Attack Surface
  - Different Vulnerabilities

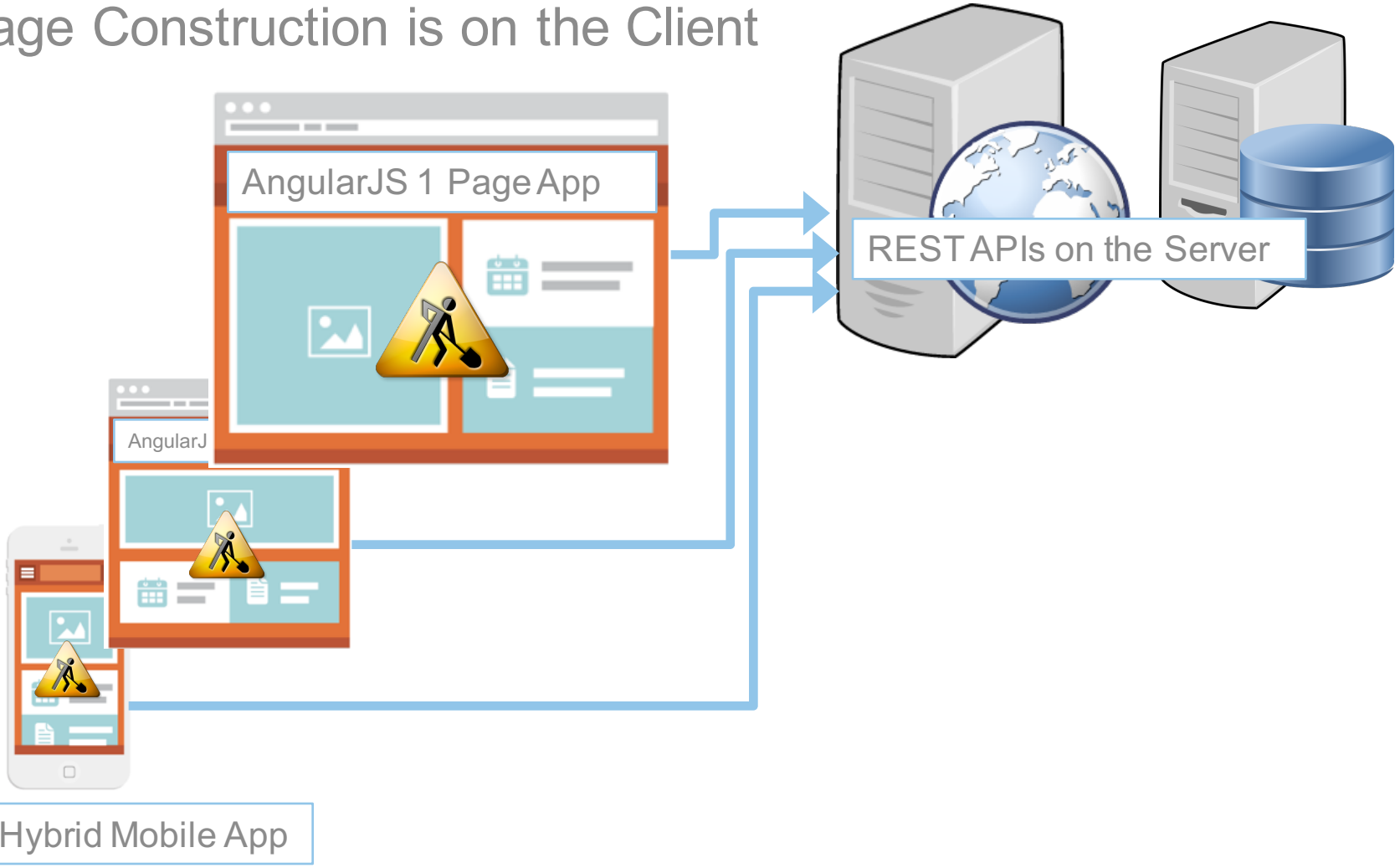
WEB APPILICATIONS ARE THIN-CLIENT



Page Construction is on the Server

MOBILE FIRST APPS ARE THICK CLIENT

Page Construction is on the Client



THIN VS THICK CLIENT REQUEST STREAMS

User Actions

1) Login
2) Click the “Send Money” button
3) Enter amount and destination where to send the money
4) Enter 2FA challenge from a txt message

Web 1.0 App Request Stream

- **(1) GET /**
  response: csrf-token
- **(1) POST /signin**
  fields: username, password, csrf-token
- **(2) GET /spend**
  response: csrf2
- **(3) POST /spend-action**
  fields: destination, amount, csrf2
  side-effect: sends txt to registered phone
- **(4) POST /validate-mfa**
  fields: mfa-value, csrf2

Mobile First App Request Stream

- **(1) GET /**
- **(1) GET /signin**
- **(1) GET /angular-templates**
- **(1) POST /api/signin**
  headers: <ddos and reply prevention>
  json: { username, password }
- **(4) POST /api/<customerId>/mfa-challenge**
  headers: X-App-Customer, X-App-Session
  json: { transaction: “spend”… }
- **(3) POST /api/<customerId>/spend**
  headers: X-App-Customer, X-App-Session
  json: { dest, amt, mfa-value …}

RESPONSE DATA – THE ACHILLES HEAL

Request

```
PUT /api/v2/customers/signin HTTP/1.1
Host: www.blah.com
X-Customer-Session: e3934204bd1e2444705864b830ec961b33914716
X-Customer-Id: xxx457
X-App-Id: angularjs
X-App-Version: 27aace2e0d
Connection: keep-alive

{"action":"signin",”username”, “foo@bar.com”, “password”: “…”}
```

Response

```
HTTP/1.1 200 OK
Server: blah-nginx
Content-Type: application/json; charset=utf-8

{"response":{"status":{"code":0,"customerState":{"isEmailVerified":true,"isMfaVerified":true,"custo
merStatus":2}},"customer":{"id":xxx457,"email":”foob@bar.com","currencyLocale":"en",…
"featureSwitches":{"ENABLE_USD":true},"mfaBuyMinimum":0,"mfaSellMinimum":0,"mfaSpendMinimum":0,"mfa
Mechanism":0,"customerStatus":2,"customerState":"active","exchangeRate":{"USD":{"base":"BTC","quote
":"USD","rate":240.75,"timestamp":1444051295583}},…
"instantAccessCap":0,"weeklyBankDepositLimit":2,"weeklyBankWithdrawLimit":2,"weeklyCreditCardDeposi
tLimit":1,"weeklyCreditCardWithdrawLimit”:3,…
"allowedCurrencies":["BTC","USD"],…
"profile":{"publicId":”2"},"exchangeRates":{"USD":{"BTC":0.004153686396677051},"BTC":{"USD":240.75}
},"baseCurrencyCode":"USD","acl":{"convertCurrency":"ENABLED","deposit":"ENABLED","linkFiatAccount"
:"ENABLED","request":"ENABLED","spend":"ENABLED","unlinkFiatAccount":"ENABLED","updateEmail":"ENABL
ED","updateCredentials":"ENABLED","updatePhone":"ENABLED","updatePii":"ENABLED","viewAccount":"ENAB
LED","withdraw":"ENABLED"},
"accounts":[{"id":nnnnnnn, …],
"customerLevel":2,"fullName":”…"},
"fiatAccounts":[],"trustedDeviceValue":”42"}}
```

CIRCLE

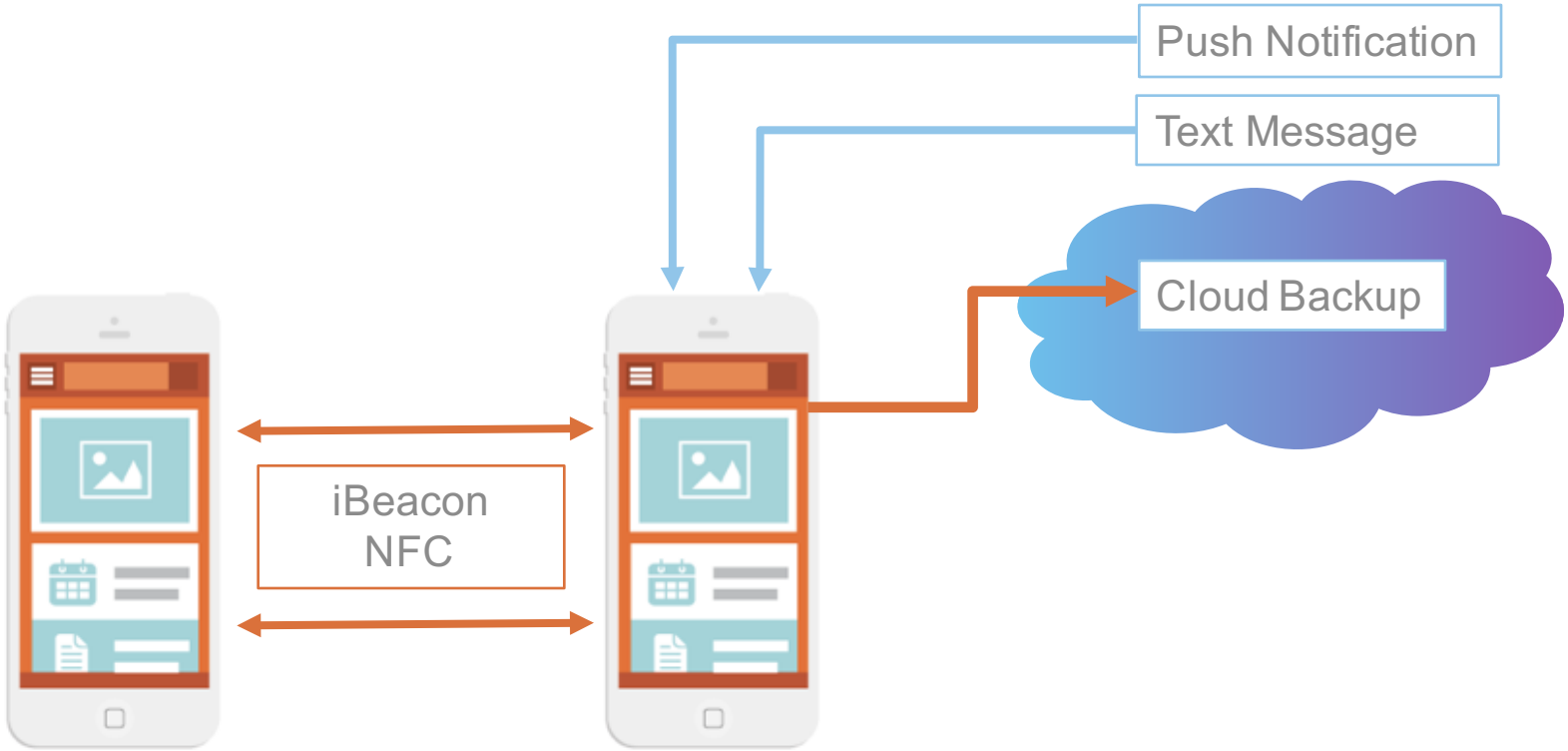
OWASP TOP 10 IMPLICATIONS

A1-Injection	No difference between Mobile First and Web applications.
A2-Broken AuthN and Session Mgmt	(Stolen) Mobile devices are a challenge for protecting the session ID. Future capabilities on the device will make mobile apps better than mobile web apps.
A3-XSS	Reflected XSS is almost a non-issue on Mobile First apps.
A4-Insecure Direct Object References	Reduced in Mobile First apps because more state is held in the client.
A5-Security Misconfiguration	No difference between Mobile First and Web applications.
A6-Sensitive Data Exposure	Mobile First apps will be more likely to have client-side trust and sensitive data exposure problems.
A7-Missing Function Level Access Control	No difference between Mobile First and Web applications, but modern technologies like Express (Node.js) make access control easier.
A8-CSRF	Custom HTTP headers is a strong control for CSRF.
A9-Using Components with Known Vulnerabilities	Change “Known” to “Unknown” for Mobile First applications.
A10-Unvalidated Redirects and Forwards	Fewer redirects and forwards in Mobile First apps.

MOBILE TOP 10 IMPLICATIONS

M1-Weak Server Side Controls	See OWASP Top 10 slide
M2-Insecure Data Storage	(Stolen) Mobile devices are a challenge for protecting the session ID. The need to stored application data locally is application dependent.
M3-Insufficient Transport Layer Protection	Mobile First apps must pin using public CA certificates because the browser validates the certificate for the API endpoint.
M4-Unintended Data Leakage	Same as A6. Mobile First apps will be more likely to have client-side trust and sensitive data exposure problems.
M5-Poor Authorization and Authentication	Mobile apps wreak havoc on 2FA schemes; especially “Lost/Reset Device”.
M6-Broken Cryptography	For Mobile First applications, the issue is more about proper key management than the “encryption process”.
M7-Client Side Injection	As written, M7 seems like a variation of M2.
M8-Security Decisions via Untrusted Inputs	As written, M8 is a variation of M3. For Mobile First, the issue is security decisions done (only) by the client.
M9-Improper Session Handling	For Mobile First, the question is how the application is managing long running sessions (not whether).
M10-Lack of Binary Protections	Future capabilities on the device will make mobile apps better than mobile web apps.

APP TO APP/APP TO CLOUD & OTHER FEATURES



CONCLUDING REMARKS

- Mobile First changes application development beyond design
- Mobile First changes the application architecture: thick client
- Mobile First apps will be hybrid
- AppSec must adapt to Mobile First: Mobile First AppSec
  - Focus in the Response data not the Request parameters
  - Look for API request sequences that leak data or should be atomic
  - Application specific HTTP headers are your friend
  - The API server isn't the only thing that the mobile app communicates with

CIRCLE

THANKS FOR LISTENING

Questions?