Who, What, Where, How: Five Big Questions in Mobile Security

Jacob West
CTO, Fortify Products
HP Enterprise Security

Session ID: ASEC-R31
Session Classification: Intermediate

Why is mobile security an imperative?

Who will be held accountable?

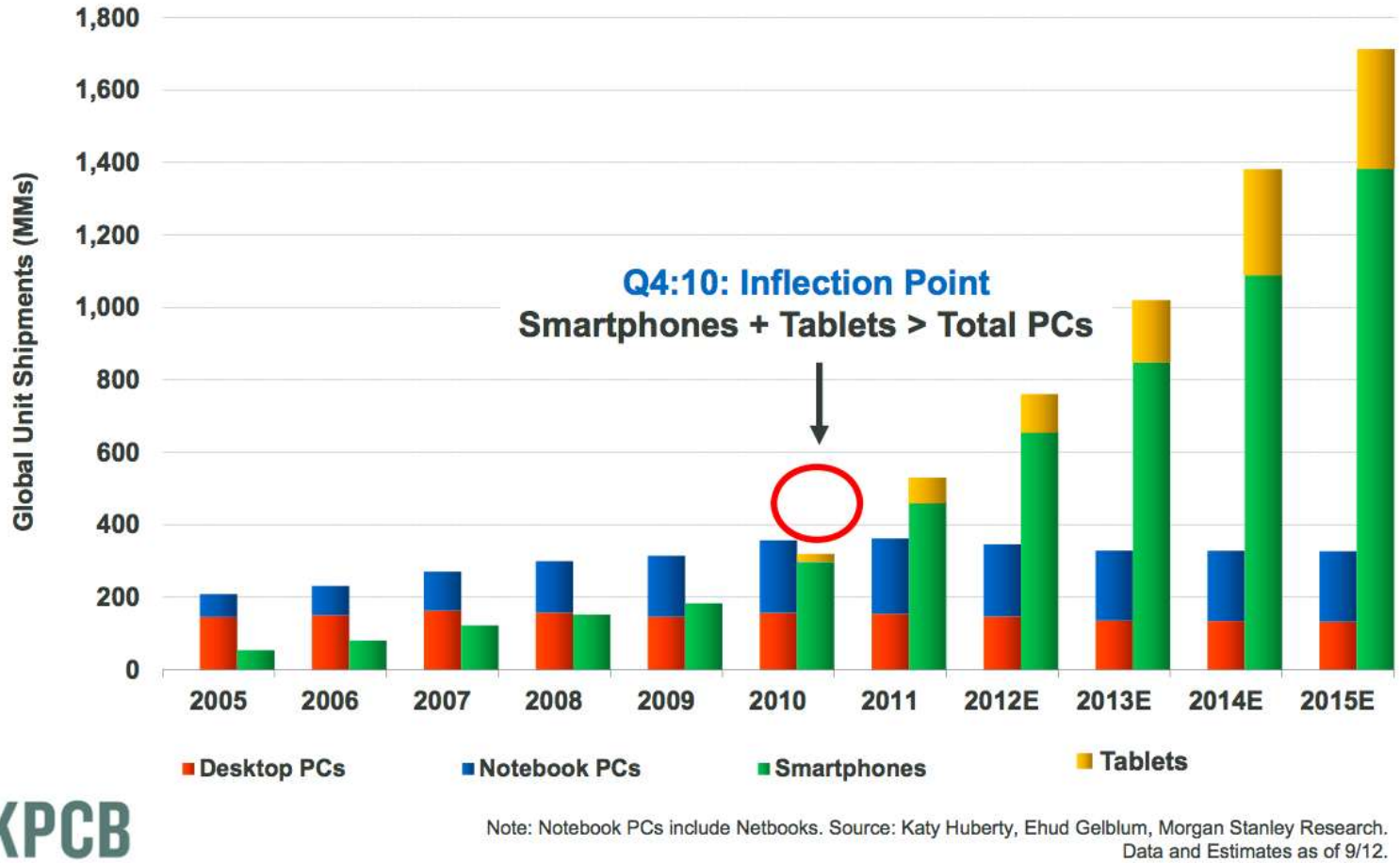
What platform strategy makes sense?

Where are mobile apps developed?

How do we build secure mobile apps?

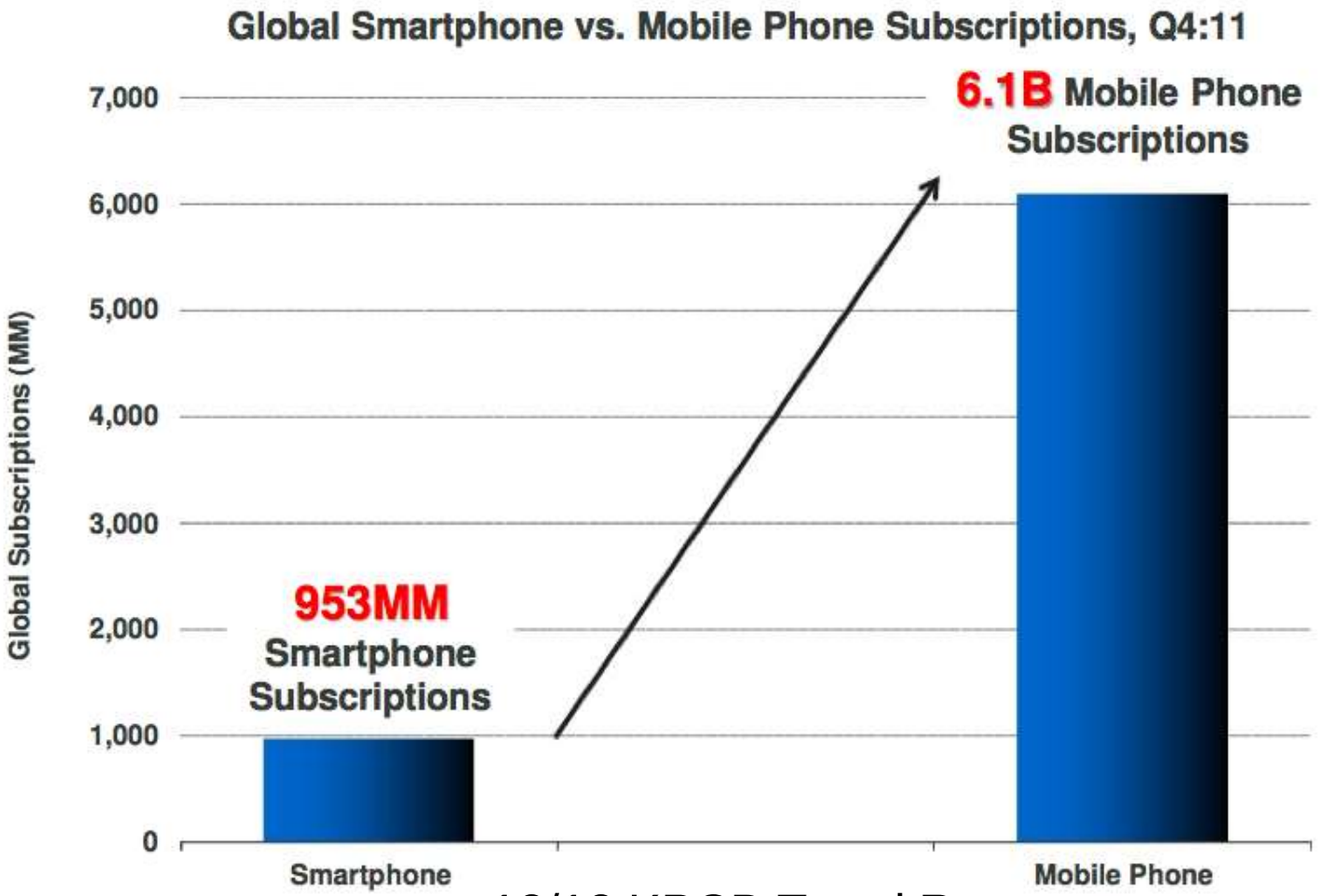
Why is mobile security an imperative?

Mobile Devices are Taking Over



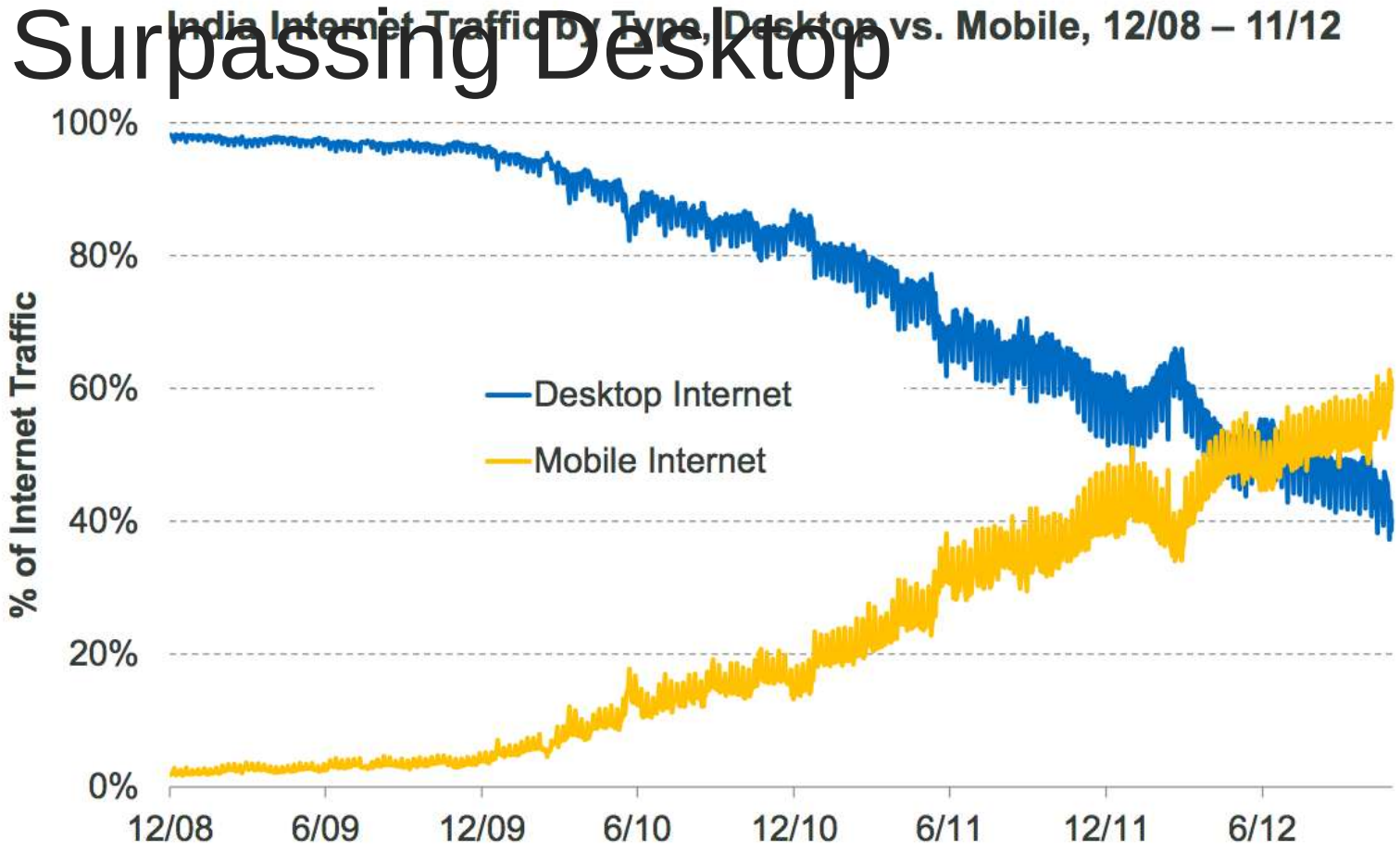
12/12 KPCB Trend Report

With Lots More to Come



12/12 KPCB Trend Report

Mobile Internet Usage Surpassing Desktop



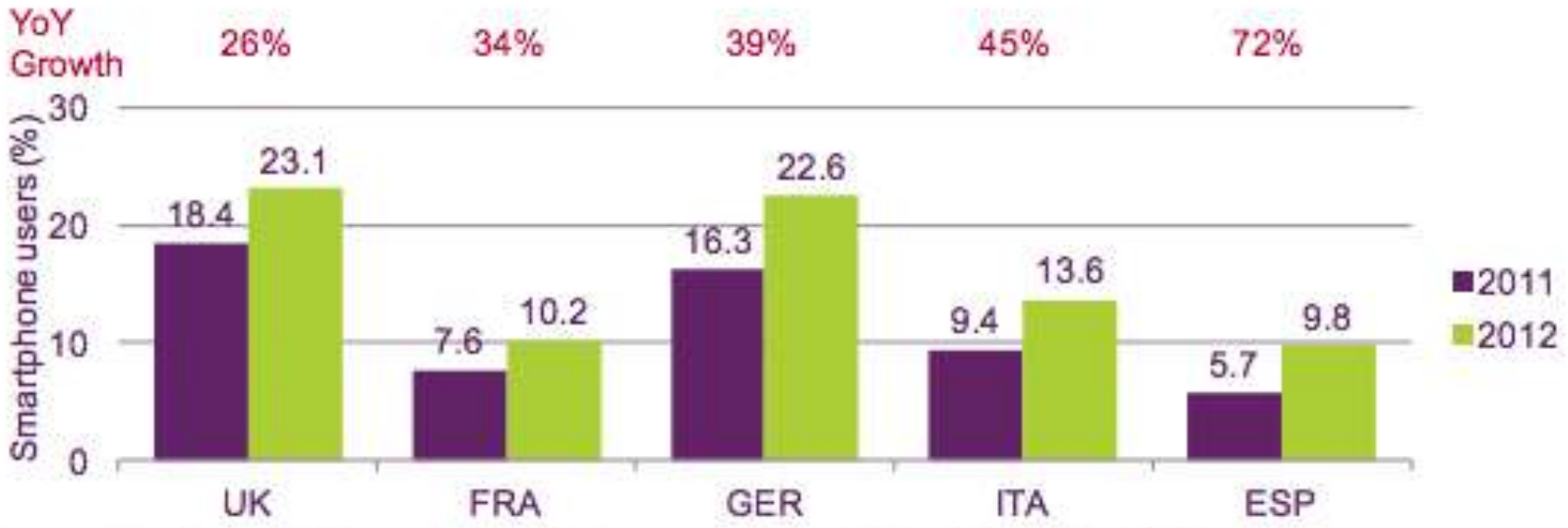
KPCB

12/12 KPCB Trend Report

Source: StatCounter Global Stats, 11/12

Smartphone Users are Shopping

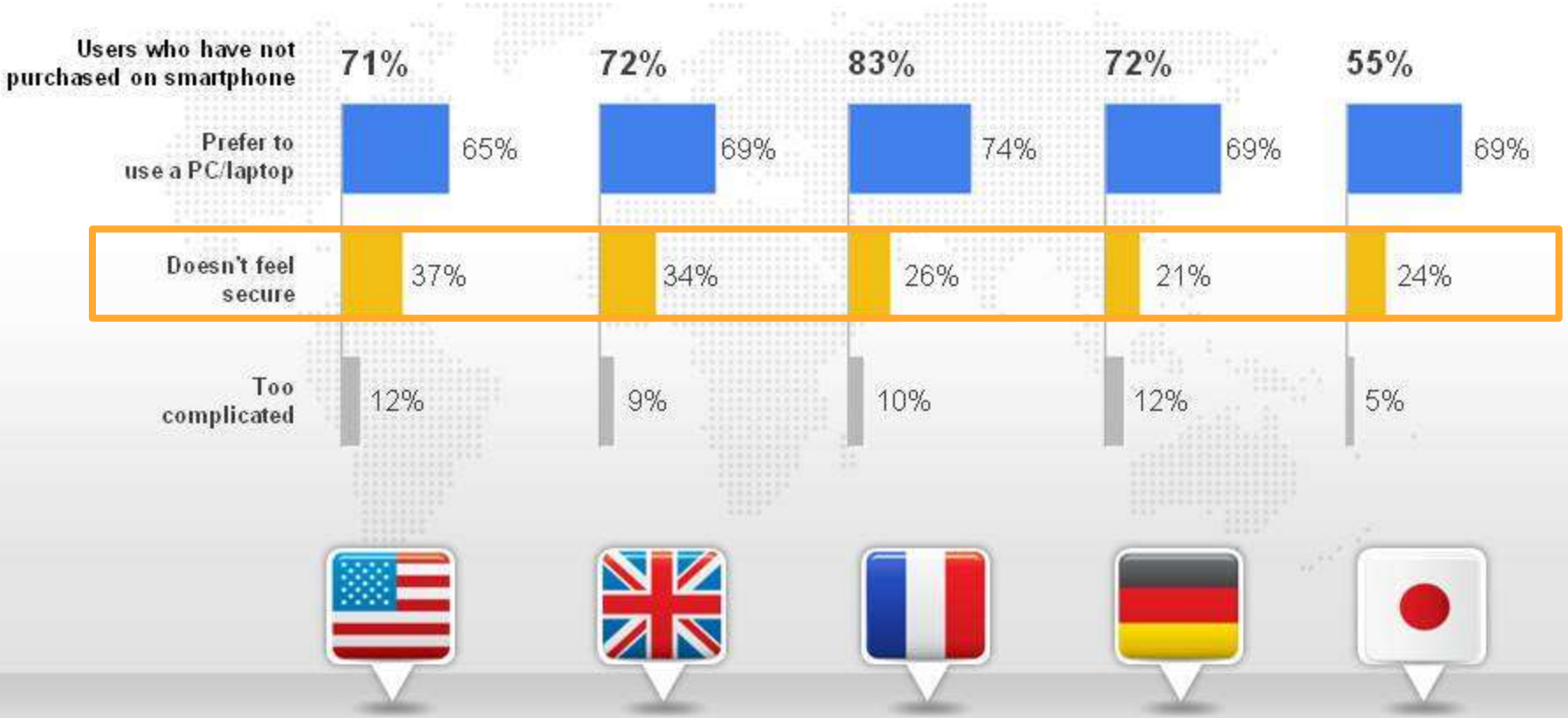
Figure 5.30 Smartphone users accessing online shopping websites



Source: comScore MobiLens, 3 month average ending May 2011 vs May 2012

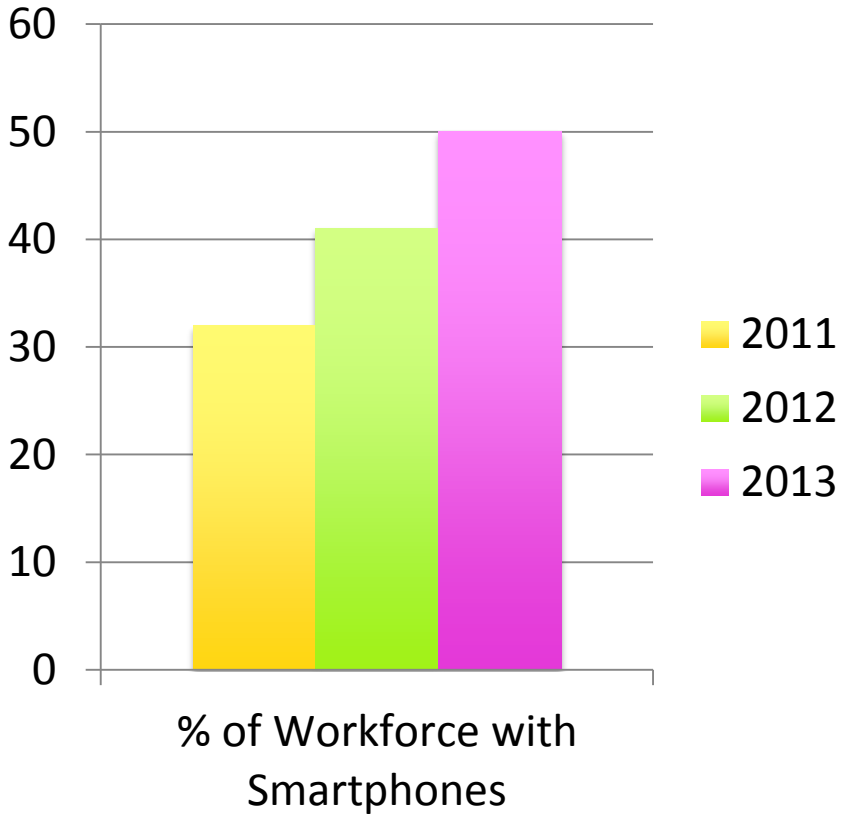
Why Mobile Users Don’t Buy

- Security is #2 reason to avoid purchases



Source: Google/MMA , Global Perspectives: The Smartphone User & Mobile Marketer, June 2011
Base: Smartphone Users (US: 6000; UK: 2000, FR: 2000; DE: 2000; JP:1000).
Base: Smartphone Users Who Have Not Made a Purchase on Device (US: 4444; UK: 1559, FR: 1653; DE: 1442; JP: 554).
Q.. Why have you not made a purchase using your smartphone?

Not Just for Consumers



- By 2016, > 50 percent of enterprise email users will rely primarily web or mobile.

  – Gartner 12/11

- Smartphones and tablets are more than 90 percent of the new device adoption.

  – Gartner 12/11

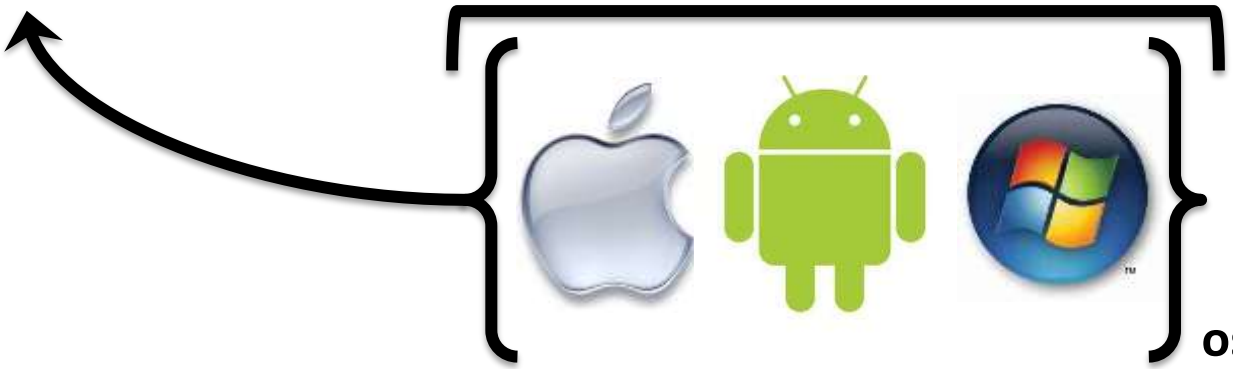
Who will be held accountable?

What *is* Mobile?

device

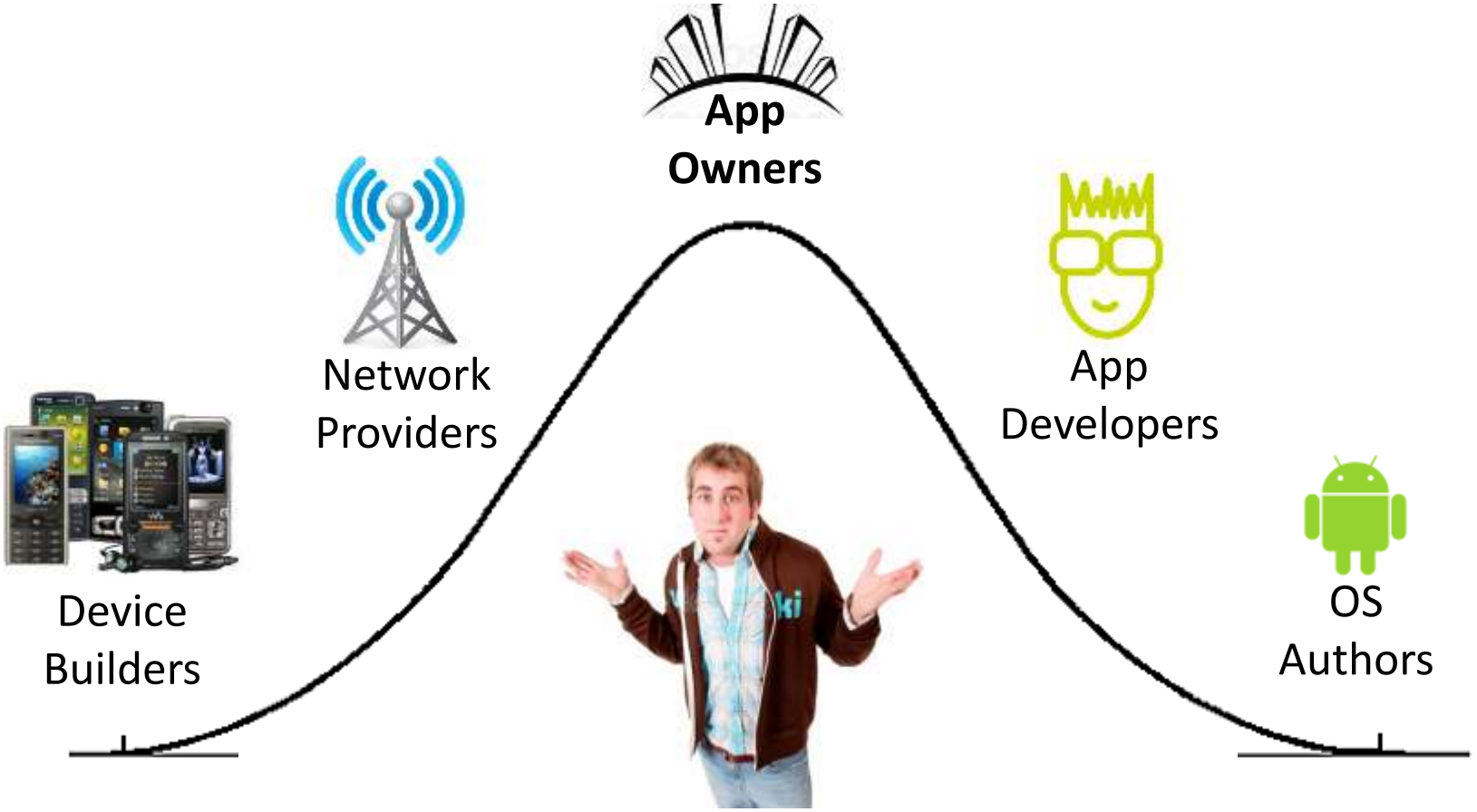
server

apps



os

Who Cares?



Who Will Users Hold Accountable?

Lots to Lose

Device Builders

- Big price tag
- *Infrequent purchase*
- Brand loyalty

Network Providers

- Big price tag
- *Monthly fee*
- Brand loyalty

Blame Game

App Owners

- Big brand impact
- Compliance
- Maintenance costs

App Developers

- No brand impact
- No compliance
- Ever more contracts

Decisions to Make

OS Authors

- Big risk, big reward
- Tied to delivery
- Developers versus users

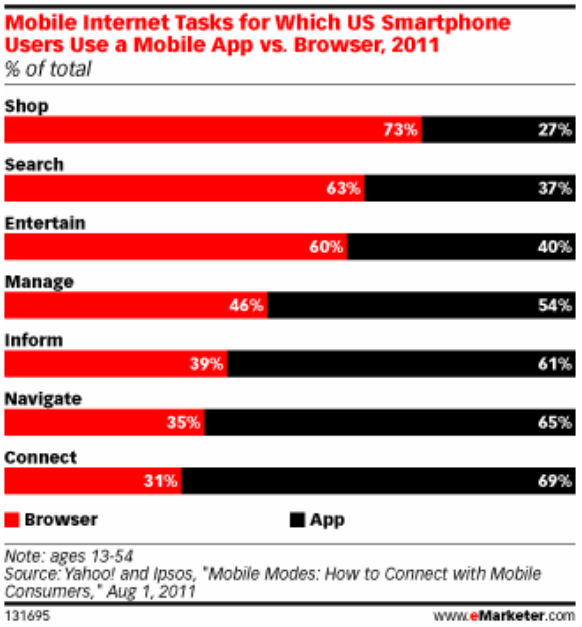
What platform strategy makes sense?

Platform Tradeoffs

- Web, native, hybrid
- Operating systems
- Developer support
- Application delivery
- Programming language

Web Versus Native

- Native mobile applications
  - Persistent on phone
  - Deeper hardware support
  - More flexlble user experience
- Mobile-optimized web apps
  - Lightweight footprint
  - Easy cross-platform model
  - Easy migration from legacy apps
- Hybrid?
  - Native container for web content
  - Cross-compiled native apps



80% by 2015
– Gartner 11/12

Working with Mobile Operating Systems

Google

BlackBerry

- Benefit of hindsight
- Security features
  - Read-only stack
  - Data encryption
  - Permissions
- Confusing
  - Wait, permissions?

Mobile OS Features: Can't We All Get Along?

- Formal communication
  - Inter-application
  - Intra-application
  - With the OS
- Platform differentiator
- A new trust boundary

Application Delivery

- Open app store model (Google Marketplace)
  - Enterprises stand-up their own app stores
  - Security can become an app-store differentiator
  - Researchers have better access
- Closed app store model (Apple App Store)
  - App store owner has much greater control
  - Victim exposure minimized with revocation capability
  - Compromise: Apple's iOS Developer Enterprise Program

Native Programming Languages

- Objective-C
  - Little-known until iOS
  - ‘Unsafe’ language makes buffer overflows a big problem
  - Limited tool support

- Java
  - Widely-known by enterprise developers
  - ‘Safe’ means no more buffer overflows
  - Better tool support

Where are mobile apps developed?

Mobile Development

- In-house
- Traditional outsourcers
- Boutique mobile development firms

In-House Development

Pros

- Leverage existing security investment
- Easier integration with legacy systems
- Control over full SDLC and artifacts

Cons

- Must train resources on new technology
- Building onto old apps may add risk
- Difficult to outsource security responsibility

Traditional Outsourcers

Pros

- Working with well-known expectations
- Expand on experience from past contracts
- Influence over SDLC and deliverables (vs. boutique firms)

Cons

- Harder to find deeply specialized skillsets
- Building onto old apps may add risk
- Outsourcing security, but not accountability

Boutique Mobile Development Firms

Pros

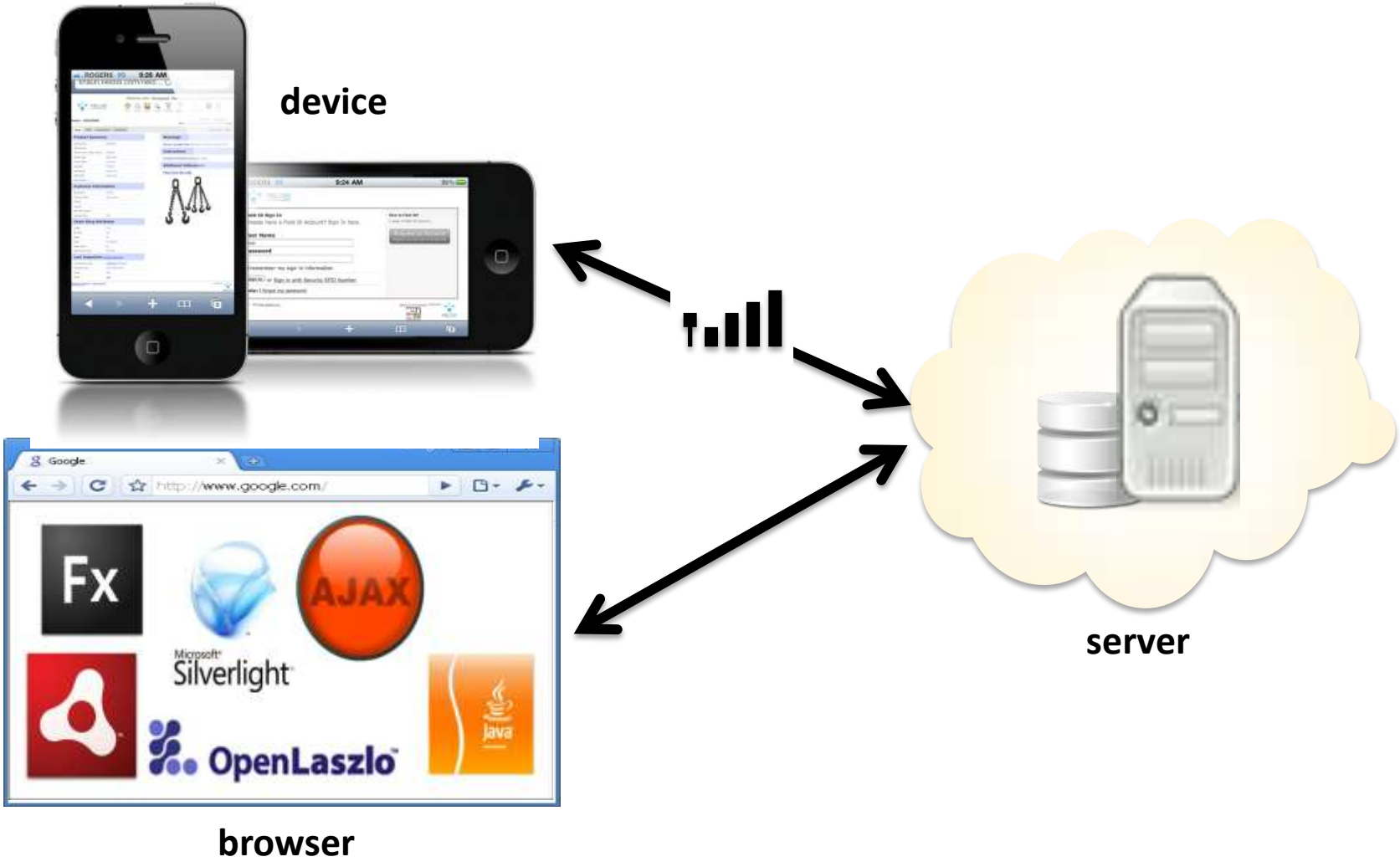
- Highly-specialized skillsets for mobile
- Opportunity to accelerate delivery
- Low-investment for high-quality result

Cons

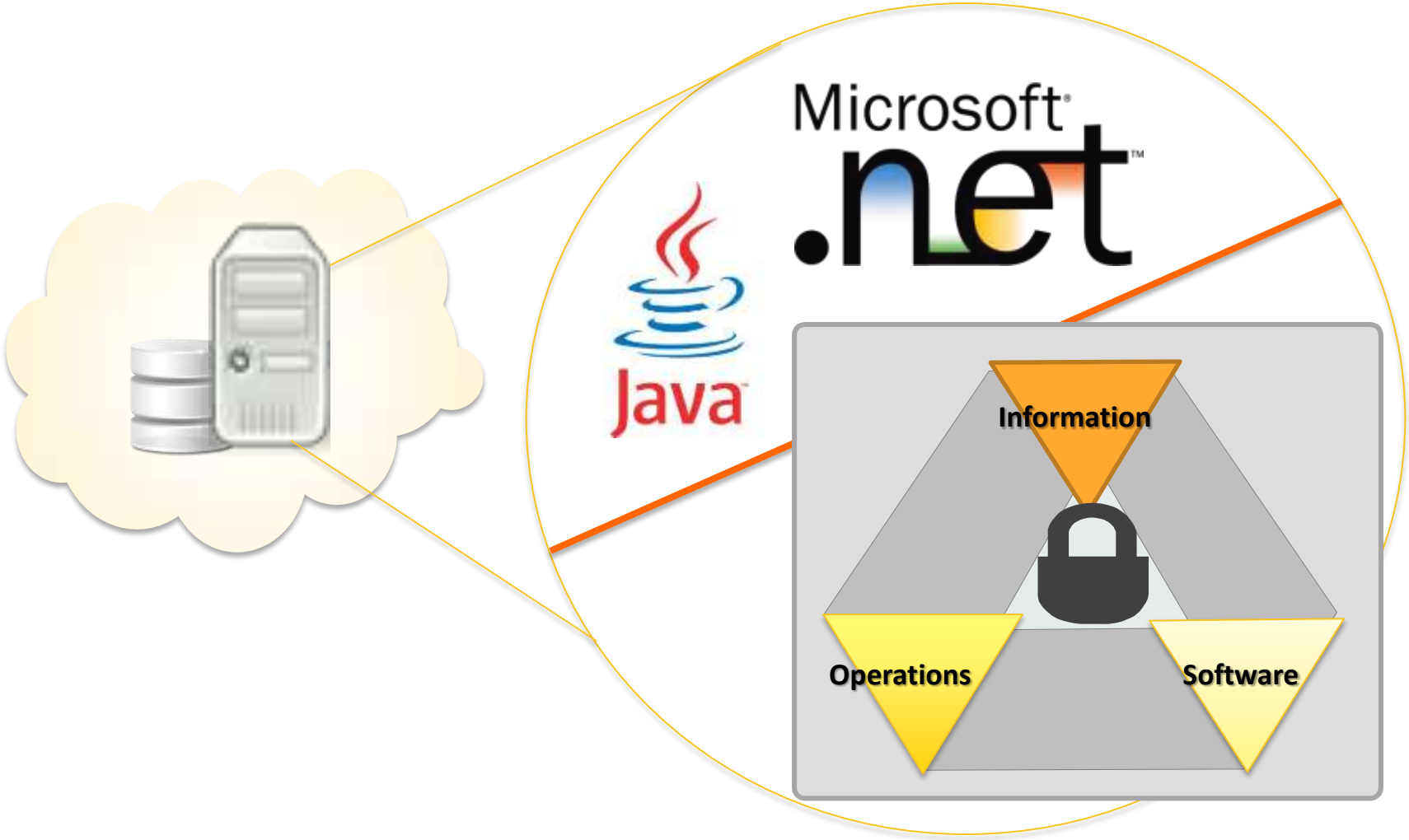
- Lack of security and engineering maturity
- Difficulty integrating with legacy systems
- Little influence over SDLC and artifacts

How do we build secure mobile apps?

Familiar Model



Same Ol' Server



Evolving Threats

Old

- Handling sensitive user and app data
- Environment and configuration
- Standbys like XSS and SQL injection

New

- Local storage (e.g. SD card)
- Communication (SMS, MMS, GPS)
- Security features (Privileges, crypto)

Google Android App

Intent Hijacking	Intent Spoofing	Sticky Broadcast Tampering	Insecure Storage	Insecure Network Communication	SQL Injection	Promiscuous Privileges

Description: Unencrypted channels can be intercepted by attackers sniffing network

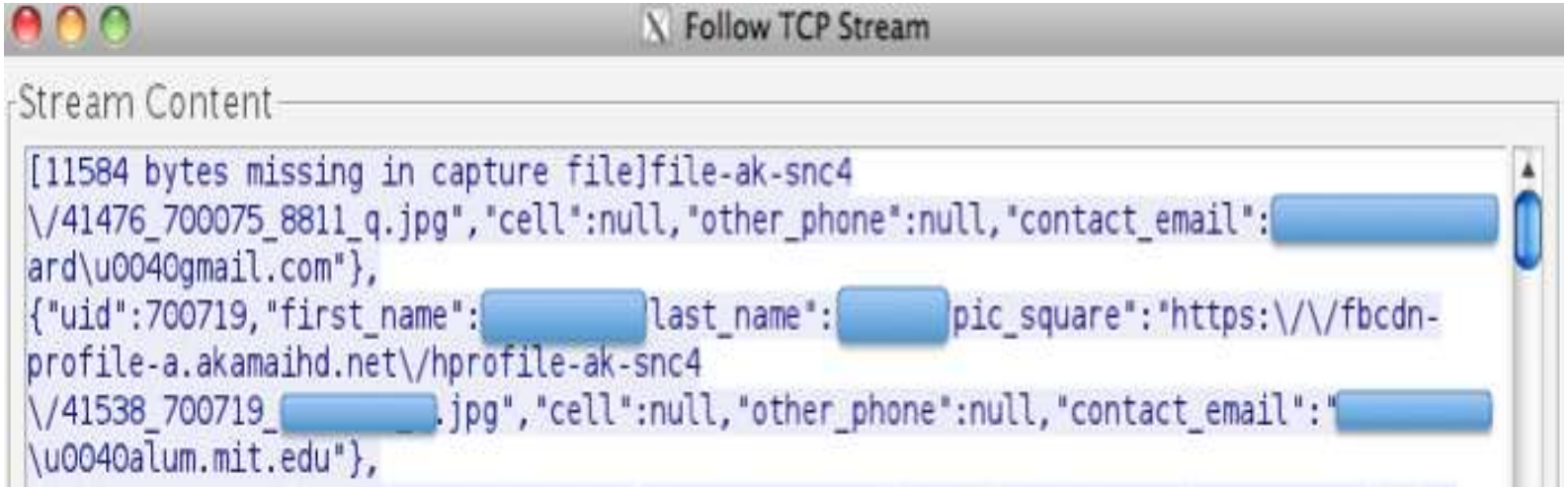
Cause: Non-HTTPS WebView connections

Fix: **Send sensitive data only over encrypted channels**

Google Android App

Intent Hijacking	Intent Spoofing	Sticky Broadcast Tampering	Insecure Storage	Insecure Network Communication	SQL Injection	Promiscuous Privileges

Facebook: Despite 'fully encrypted' option on the Web, mobile app sends in the clear

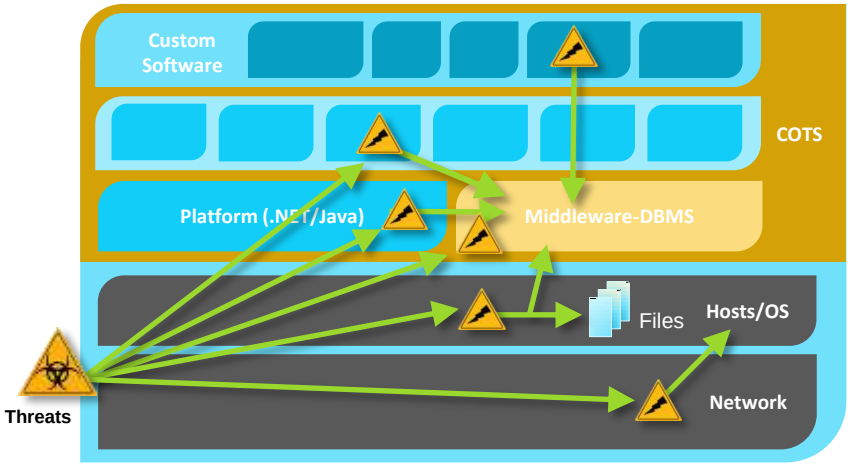


Challenges for Organizations

1

Immediate – Find & Fix

Find and **Fix** today's software vulnerabilities putting us at risk



2

Systemic

Make sure that security is built into tomorrow's software

Today: Expensive and Reactive

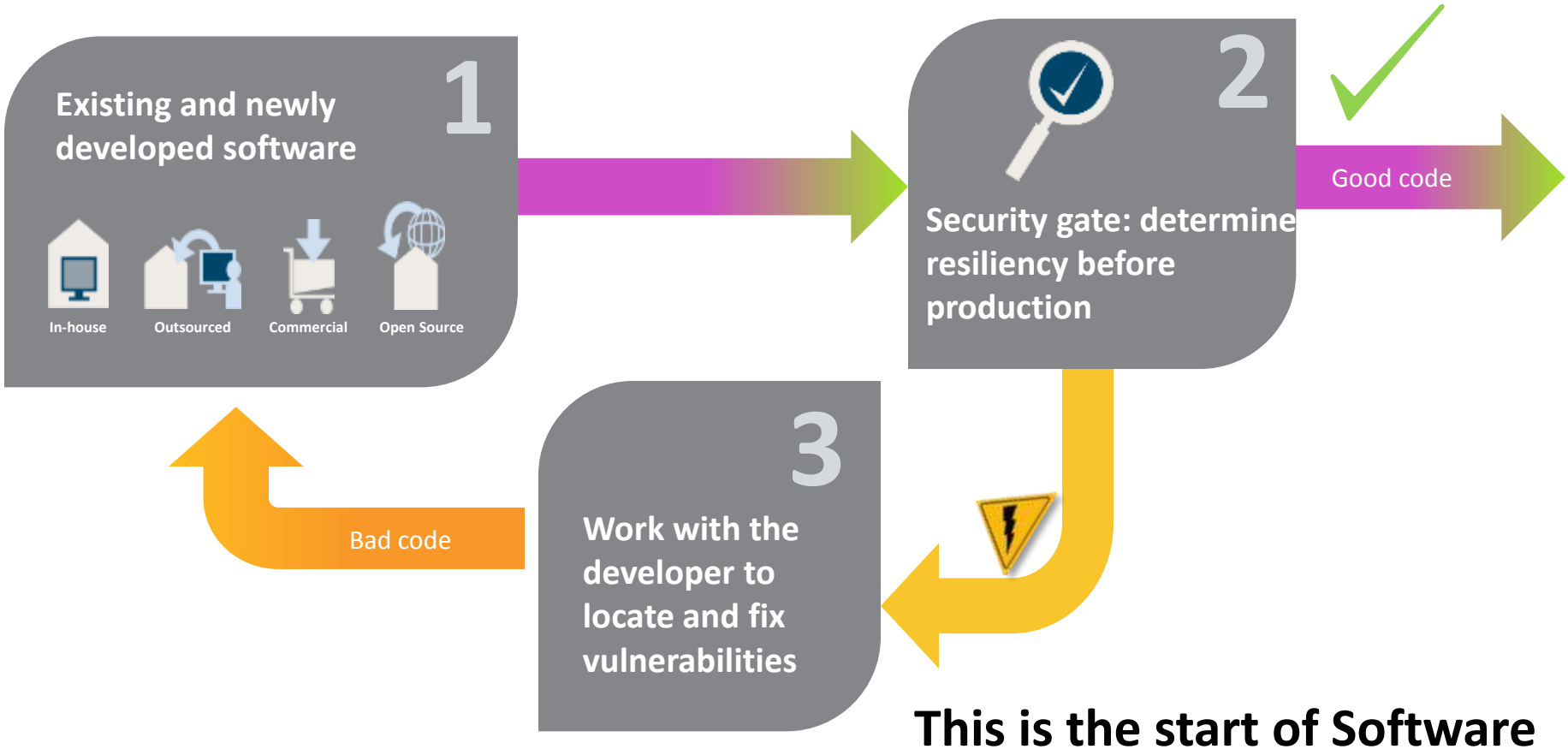
1 Somebody builds bad software

2 IT deploys the bad software

3 Breach or pen test proves our code is bad

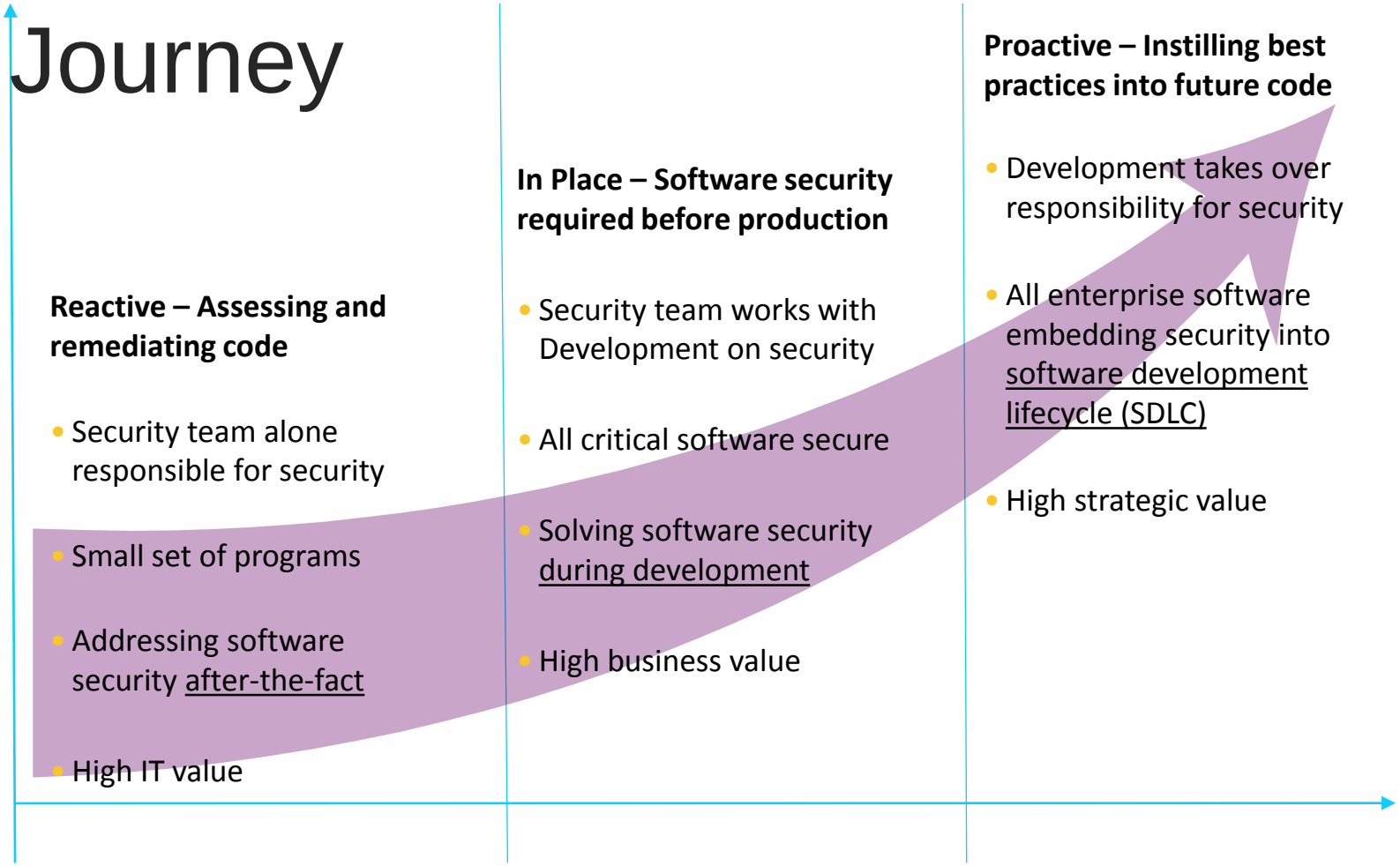
4 We convince and pay developers to fix it

A Safer, More Effective Approach



This is the start of Software Security Assurance (SSA)

Software Security Assurance Journey



Inspiration from the Industry: BSIMM4

- Real data from (51) real initiatives
- 95 measurements
- 13 repeat measurements
- McGraw, Migues, & West

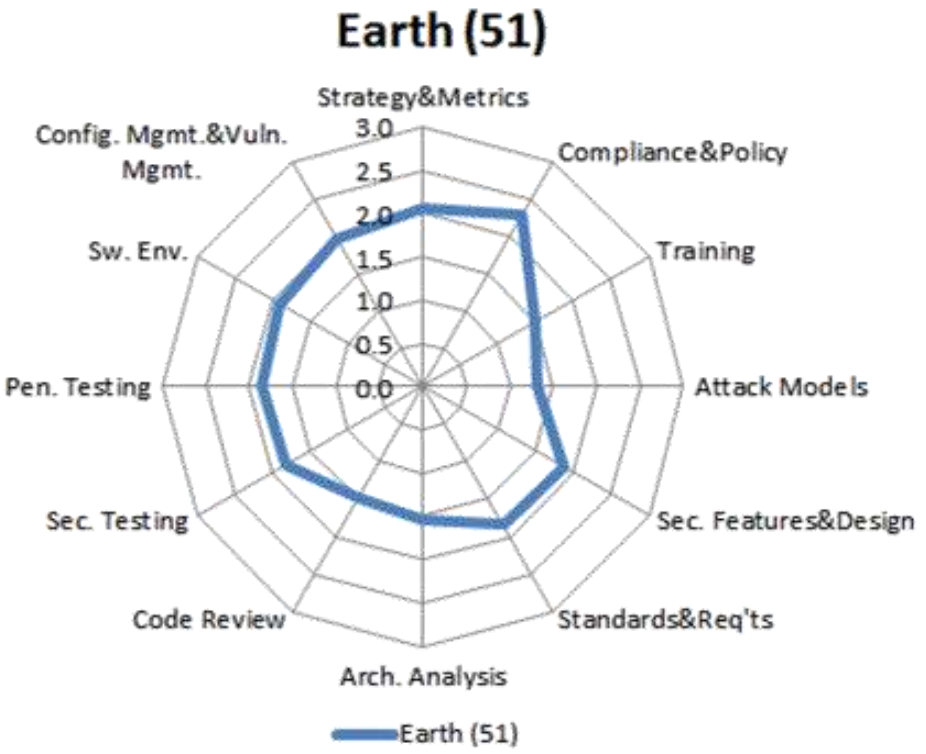
www.bsimm.com

BSIMM4: Participants

Plus 17 firms that remain anonymous

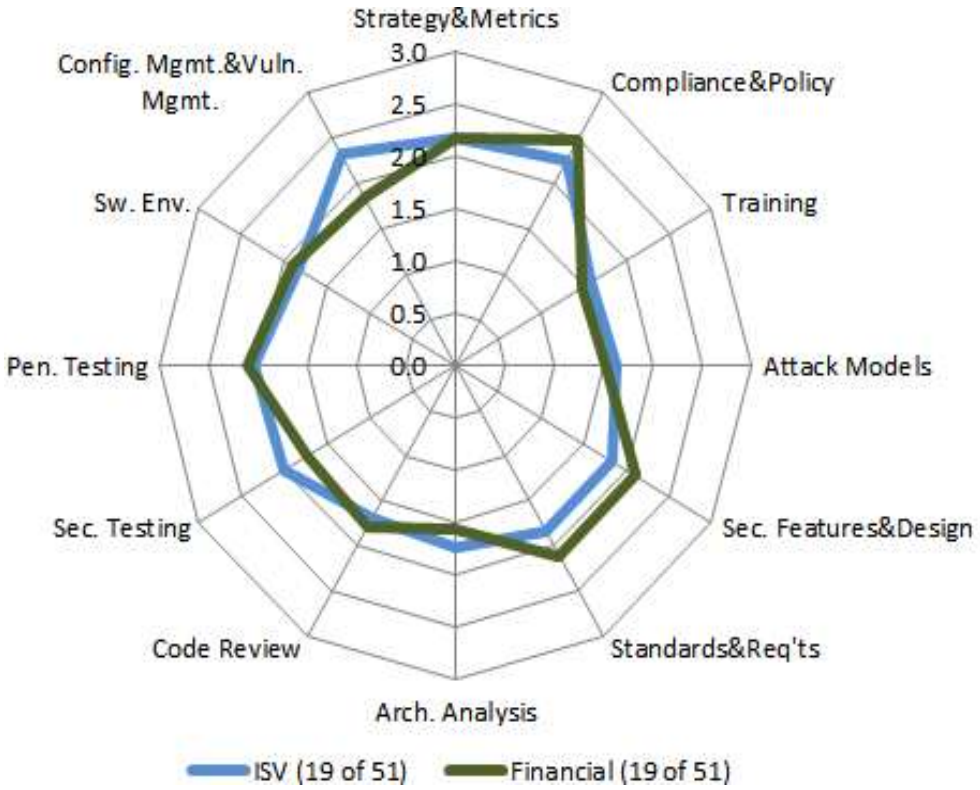
Common Activities

- Identify gates
- Know PII obligations
- Awareness training
- Data classification
- Identify features
- Security standards
- Review security features
- **Static analysis tool**
- QA boundary testing
- **External pen testers**
- Good network security
- Close ops bugs loop



No Special Snowflakes

- ISV (19) results are similar to financial services (19)
  - Do the same things
  - Can demand the same results
  - Measurement works for all



Parting Thoughts

What Questions to Ask?

- What do your apps do and for whom?
- What platform(s) do your apps support and how?
- Who develops your apps and where?
- Is there an existing SDL for other development?
- Do you rely on platform providers or app distributors for any security assurance?
- Are mobile apps prompting back-end changes?
- Are your apps appropriately permissioned?