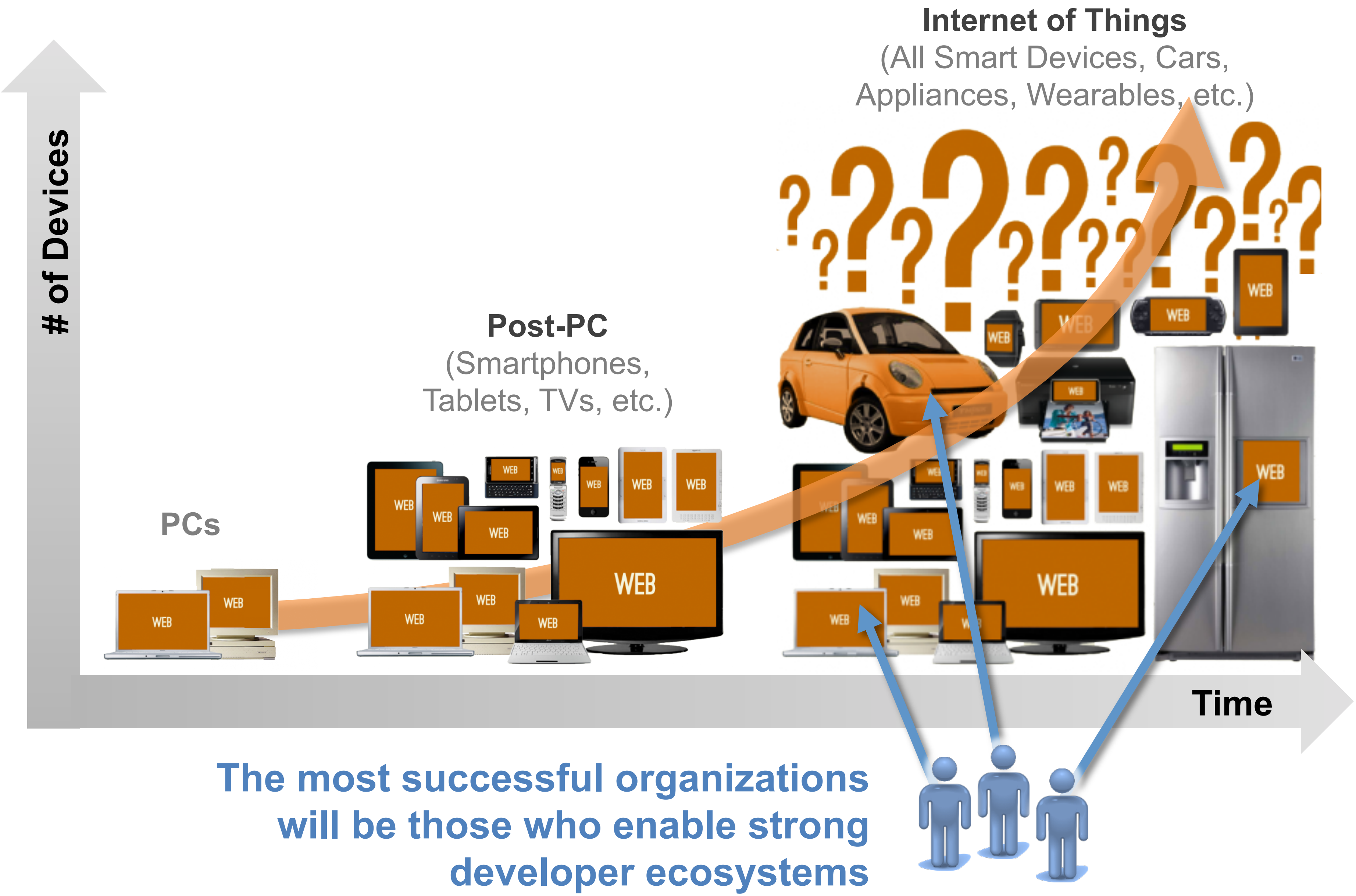
Best Practices for API Adoption

Carlo Longino
VP of Developer Program Services, WIP Factory

Richard Mendis
Chief Product and Marketing Officer, AnyPresence

Evolving Connected Technology Landscape


Internet of Things
(All Smart Devices, Cars, Appliances, Wearables, etc.)
Post-PC
(Smartphones, Tablets, TVs, etc.)
PCs
# of Devices
Time
WEB

The most successful organizations will be those who enable strong developer ecosystems

Images: Brad Frost

What do we want from our APIs?

Lower costs

Increased revenue

Faster time to market

New users

INNOVATION

Increased traffic

Stickiness/lock-in

Increased loyalty

Lead generation

Wider content spread

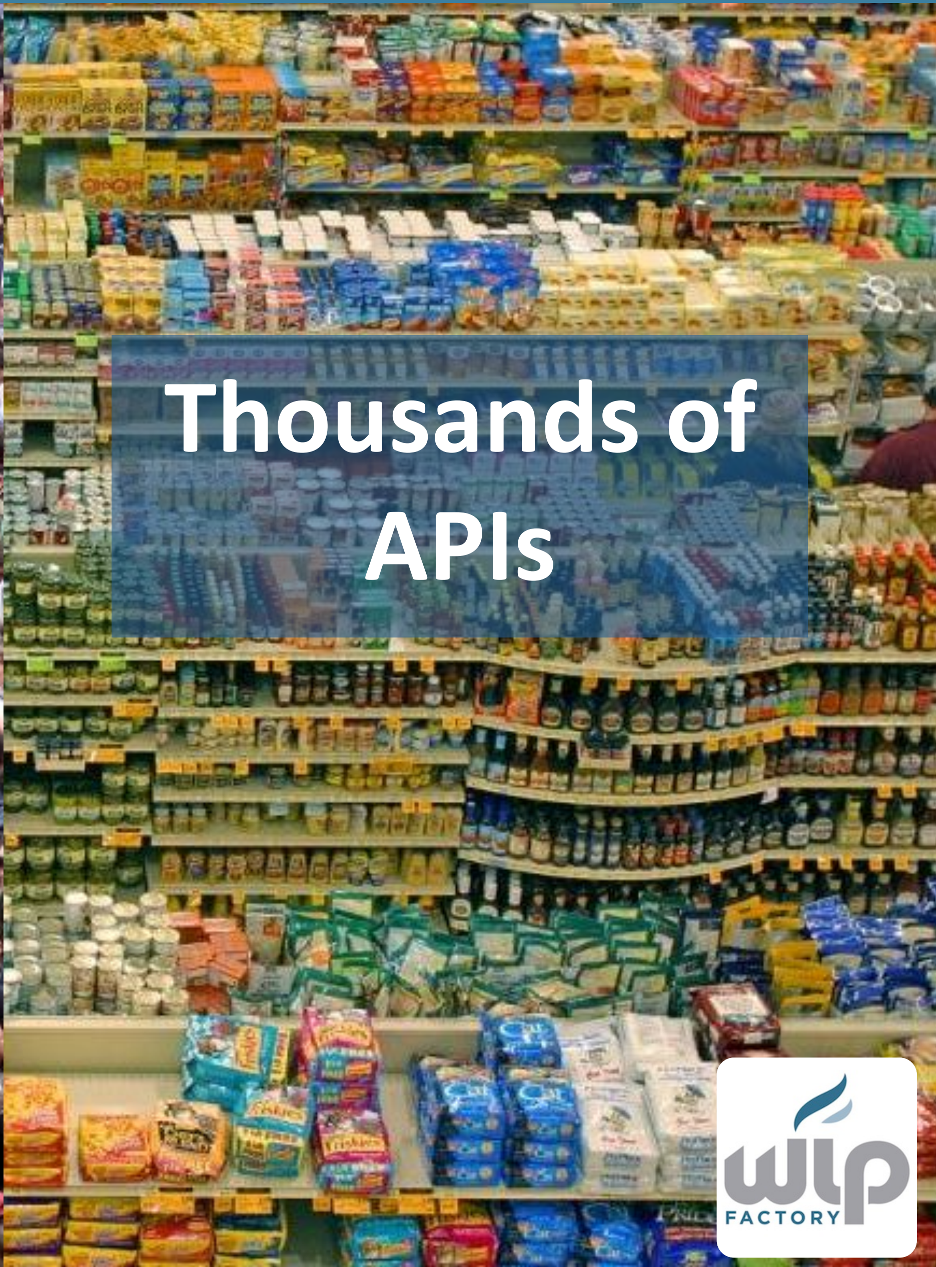
What we are really asking is:

What do we want developers to do with our APIs?

What is the landscape like?

Millions of developers

From Flickr user jamescridland



Thousands of APIs

How many developers?

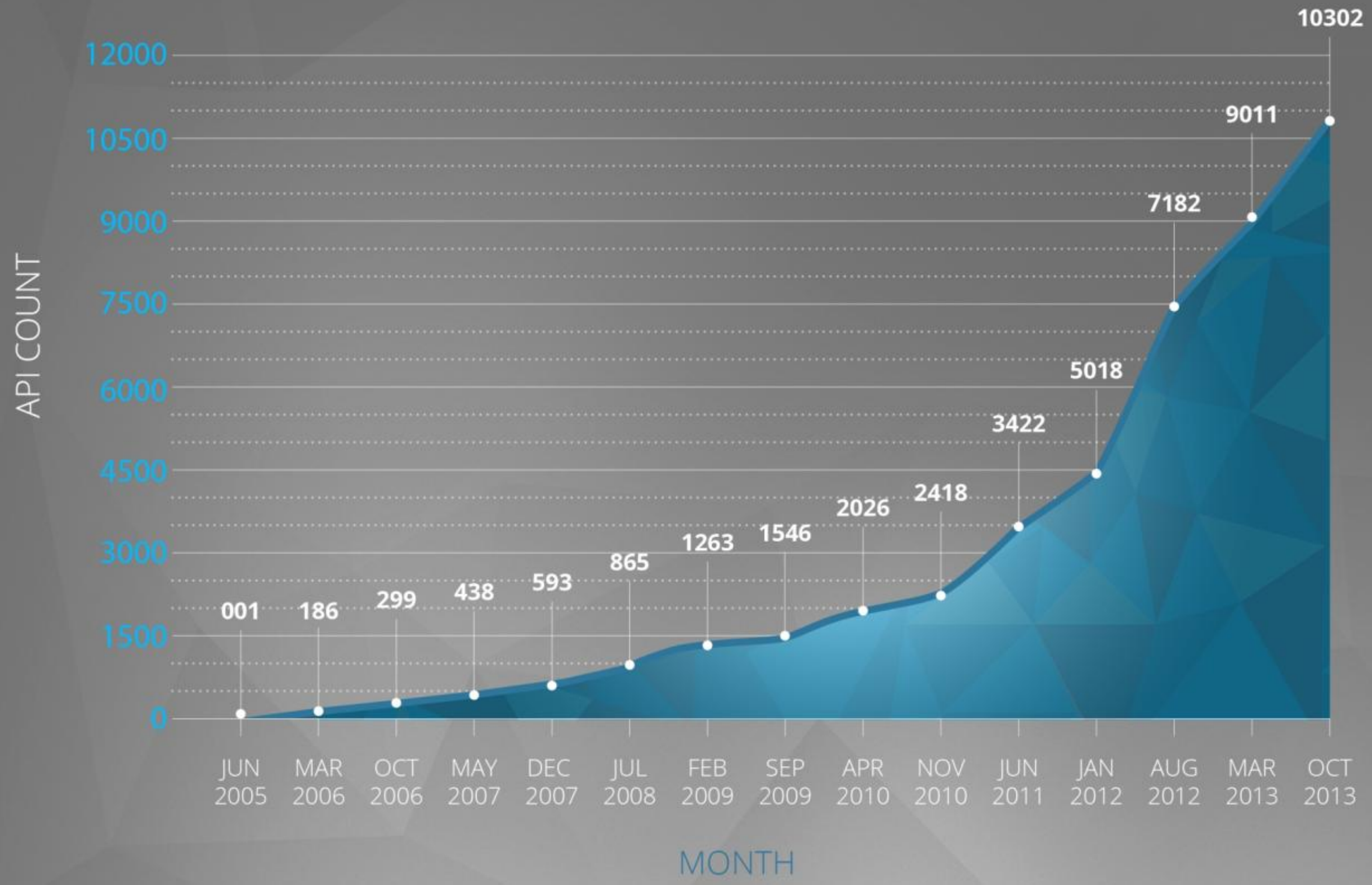
- IDC: 18.5 million Software Developers
- EDC: 18.2 million Developers
- Plumbr: 43 million Software developers
  - https://plumbr.eu/blog/how-many-java-developers-in-the-world
- Stackoverflow: 26.9 million monthly Visitors
- WIP: Over 50% of developers visit SO every week → Developers**=>50 million**

Great Math!

ProgrammableWeb

Growth In Web APIs Since 2005

This week: 12,239



wip FACTORY

What this means for your API program:

You need to focus on **the right developers**.

You need to get them to **invest their currency** – *time* – in your APIs.

Targeting “All Developers”

Targeting **“All Developers”**

is not an effective Strategy

Targeting the "Long Tail"

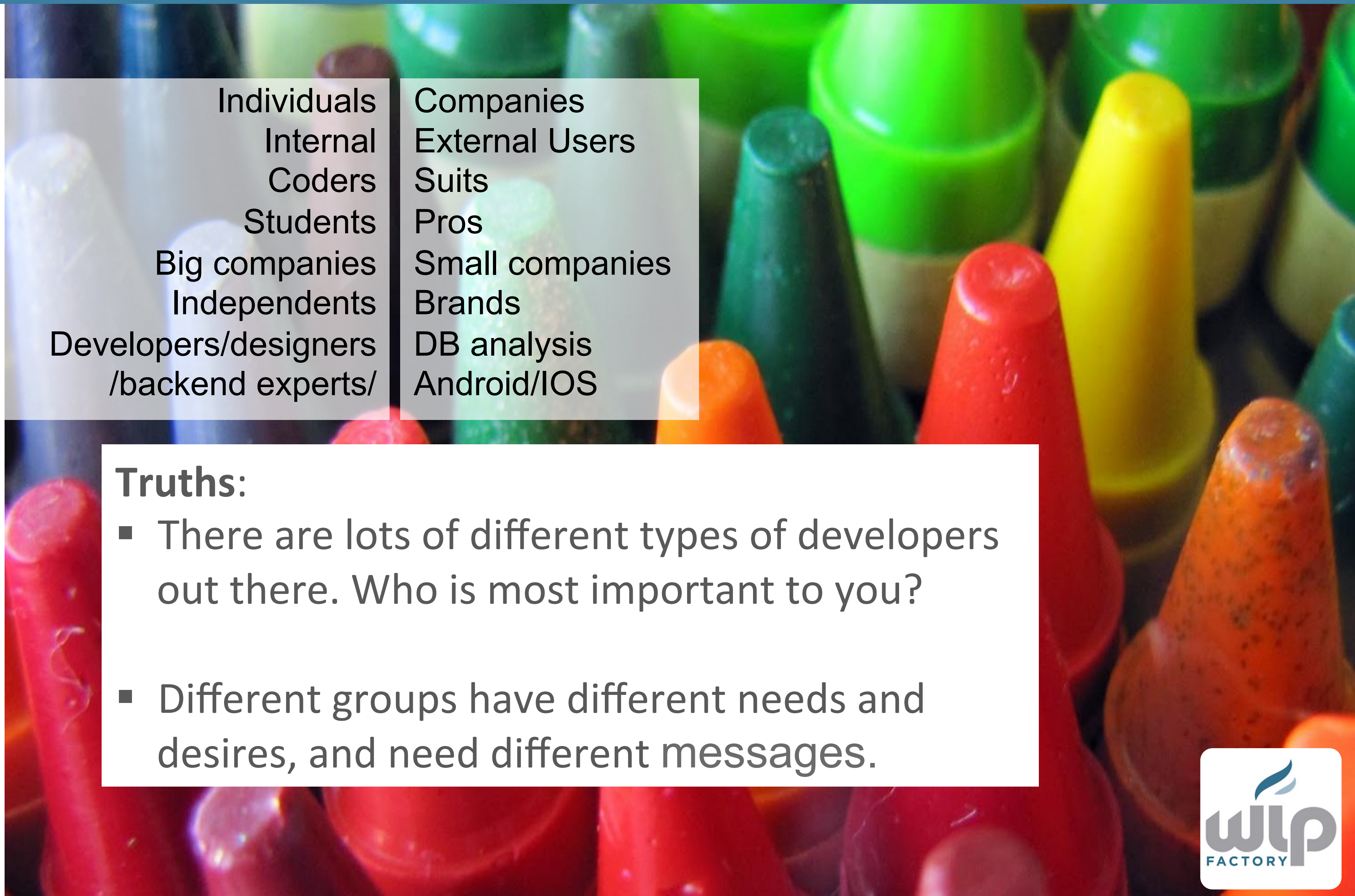
We're targeting the

"Long Tail"

That's not an effective strategy either as still too broad.

http://onebigphoto.com/a-long-tail-of-stingray/

Targeting factors



Individuals	Companies
Internal	External Users
Coders	Suits
Students	Pros
Big companies	Small companies
Independents	Brands
Developers/designers	DB analysis
/backend experts/	Android/IOS

Truths:

- There are lots of different types of developers out there. Who is most important to you?
- Different groups have different needs and desires, and need different messages.

Developers aren’t all the same

You don’t have unlimited resources.

Filter & Group based on relevant characteristics.

Each needs different services, support, marketing messages and activities.

The myth of developers

They aren’t all creative & they can’t read minds.

So help them paint the picture.

Manage YOUR expectations.

http://upload.wikimedia.org/wikipedia/commons/e/ef/Modern-Trade-Painter.jpg

Get the right group!

Once you have the right group,

Get them off to the races!

What is onboarding?

- The process of getting a developer to use your product or service
- Post-awareness, exploration, getting started, learning, going live, using your stuff

Why is it so important?

- It’s crucial to minimize the hurdles/obstacles/ friction between getting started and going live
- This is where you will lose the most developers from your program
- It’s also something you have a lot of control over!

7-Point API Onboarding Review

1 First Impression

Make it easy for the right developer to get your message and take action.

2 Product Messaging

Answer "What's in it for me?"

3 Examples and Case Studies

Show who uses your technology, what they do and how they benefit.

4 Registration

Keep the requirements light and the process quick.

5 Getting Started

Get developers up and running quickly. Minimize your TTFHW!

6 Docs and Support

Good documentation is essential – and it's part of the decision-making process.

7 Libraries and SDKs

Light the path by creating add-ons and tools that are right for your developers.

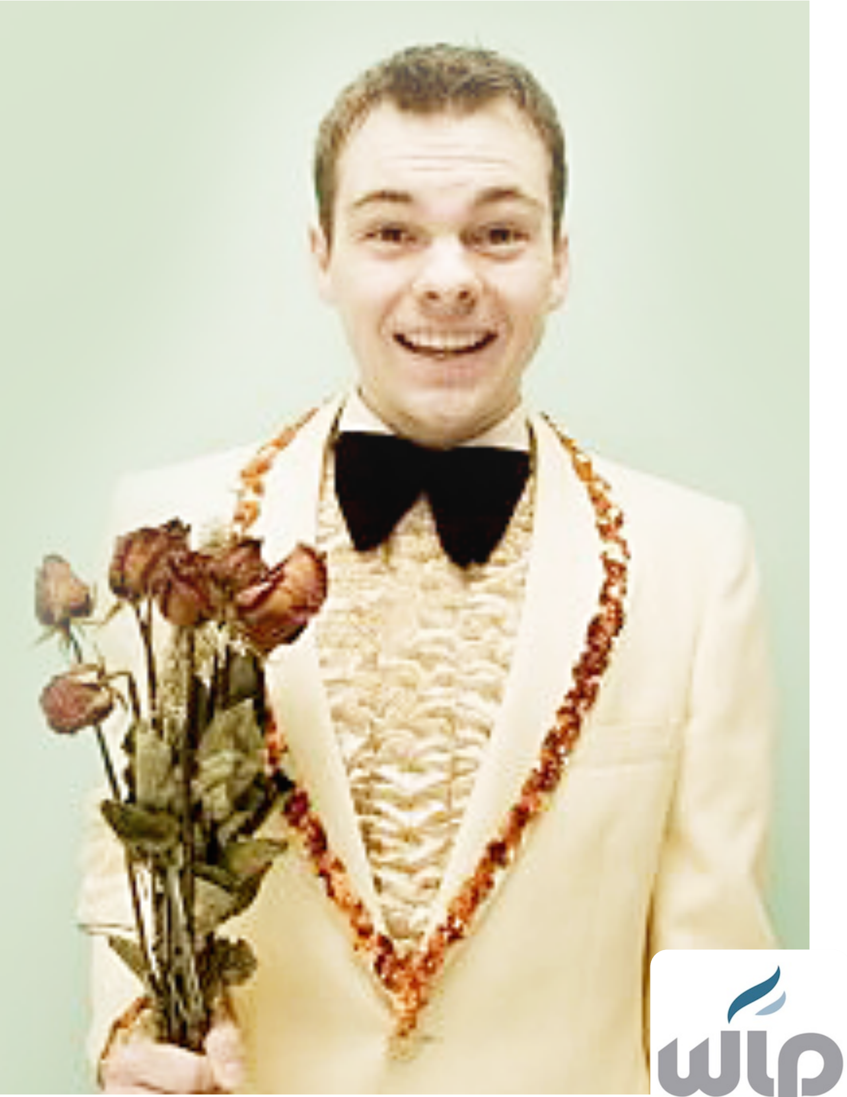
www.wipfactory.com

First Impression

1 **First Impression**

Make it easy for the right developer to get your message and take action.

- Understand WHO they are, and the timing and context around their needs.
- Who is this for?
- What does it do?
- What's in it for me?



Product Messaging

2 **Product Messaging**

Answer "What's in it for me?"

Why is it better than other solutions?

Why would I use this product?

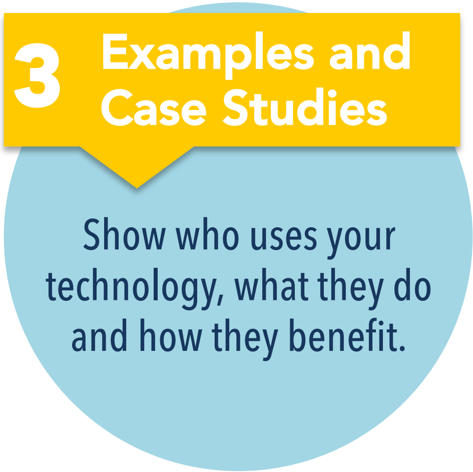
Expand on the Value Proposition

Why does it deserve my time?

Where's my win?

wip FACTORY

Examples and Case Studies



3 Examples and Case Studies

Show who uses your technology, what they do and how they benefit.

- Show developers how it’s working, rather than telling them.
- Builds understanding and credibility
- Focus on speaking to the right context
- Show the benefits

Registration

4 Registration

Keep the requirements light and the process quick.

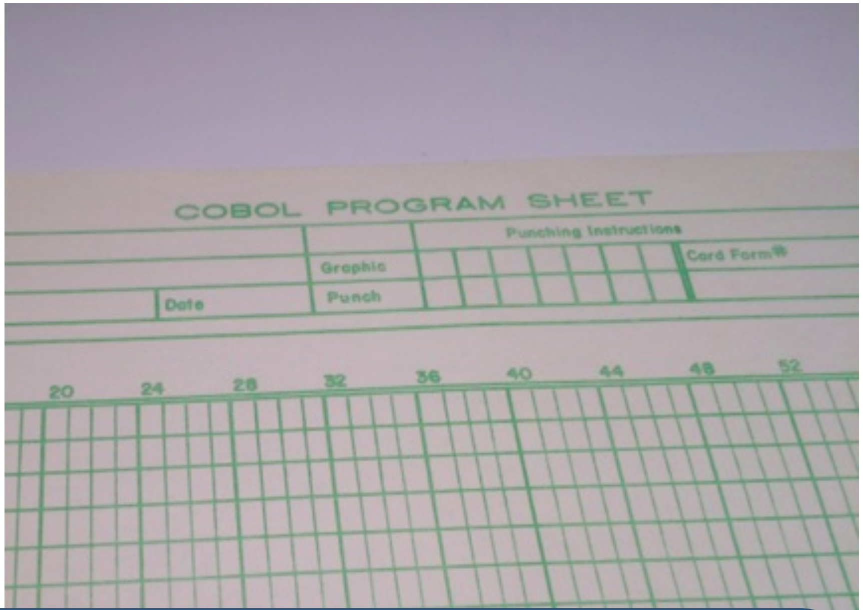
- Your best chance to screw things up!
- Ask for as little info as necessary
- **Being able to use the tool/API/SDK/ code is part of the learning and decision-making process**

Getting Started

5 Getting Started

Get developers up and running quickly. Minimize your TTFHW!

- What is your Time To First Hello World?
- How quickly can developers see results?
- Do you have a bulletproof **Quick Start Guide**?



"You'd use these to hand-write your computer programs. In pencil."

"The next day you could find out if your code compiled or not."

http://simonallardice.com/general/2014/03/17/first-computer.html

Documentation and Support

6 Docs and Support

Good documentation is essential – and it's part of the decision-making process.

github SOCIAL CODING

stackoverflow

- Get the content correct & and make it easy to use
- No PDFs, DOCs, use HTML
- Make your search work
- Don’t hide behind logins
- Don’t forget code samples, and demos
- Forums – meh… - go where developers go

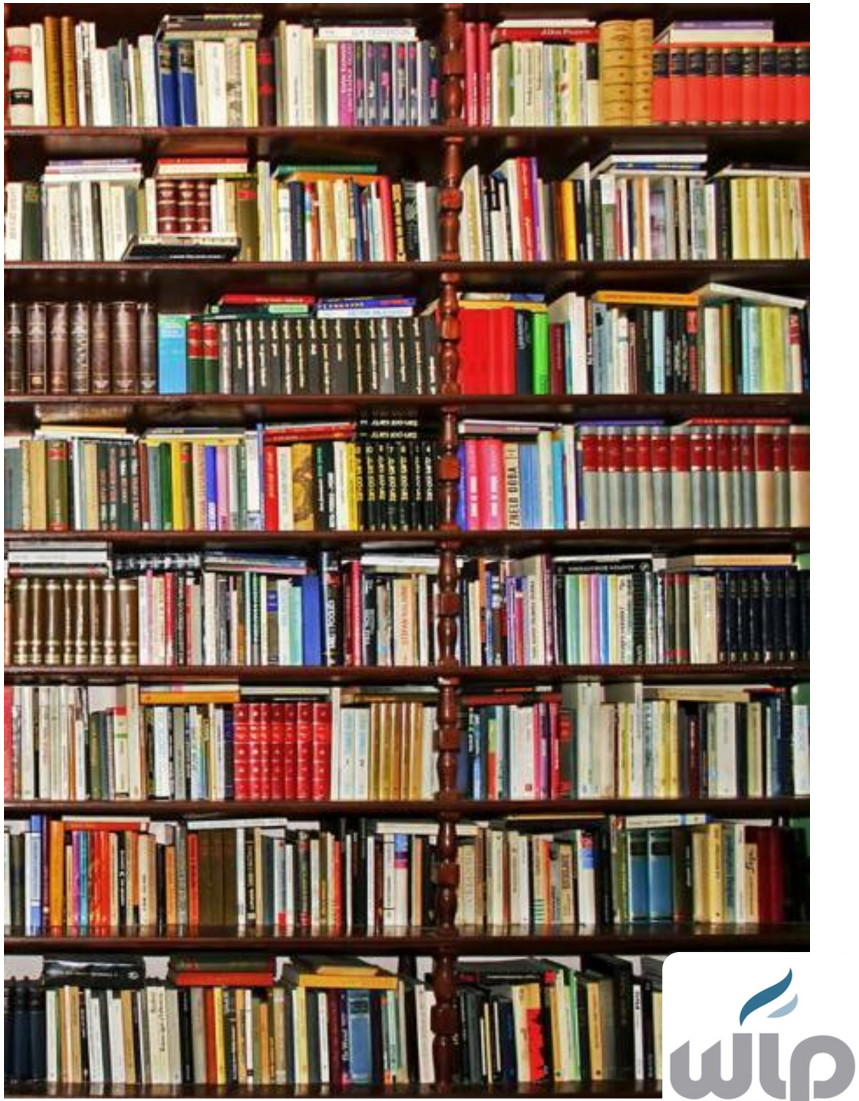
wip FACTORY

Libraries and SDKs

7 Libraries and SDKs

Light the path by creating add-ons and tools that are right for your developers.

- Create the right ones for your users
- Link to outside libraries and open-source projects
- Creates currency with community, as well as supports developers



To Sum Up:

1. Find the right developers.
2. Understand what they want and need.
3. Light the way for them, and help them down the path.
4. Manage your expectations.

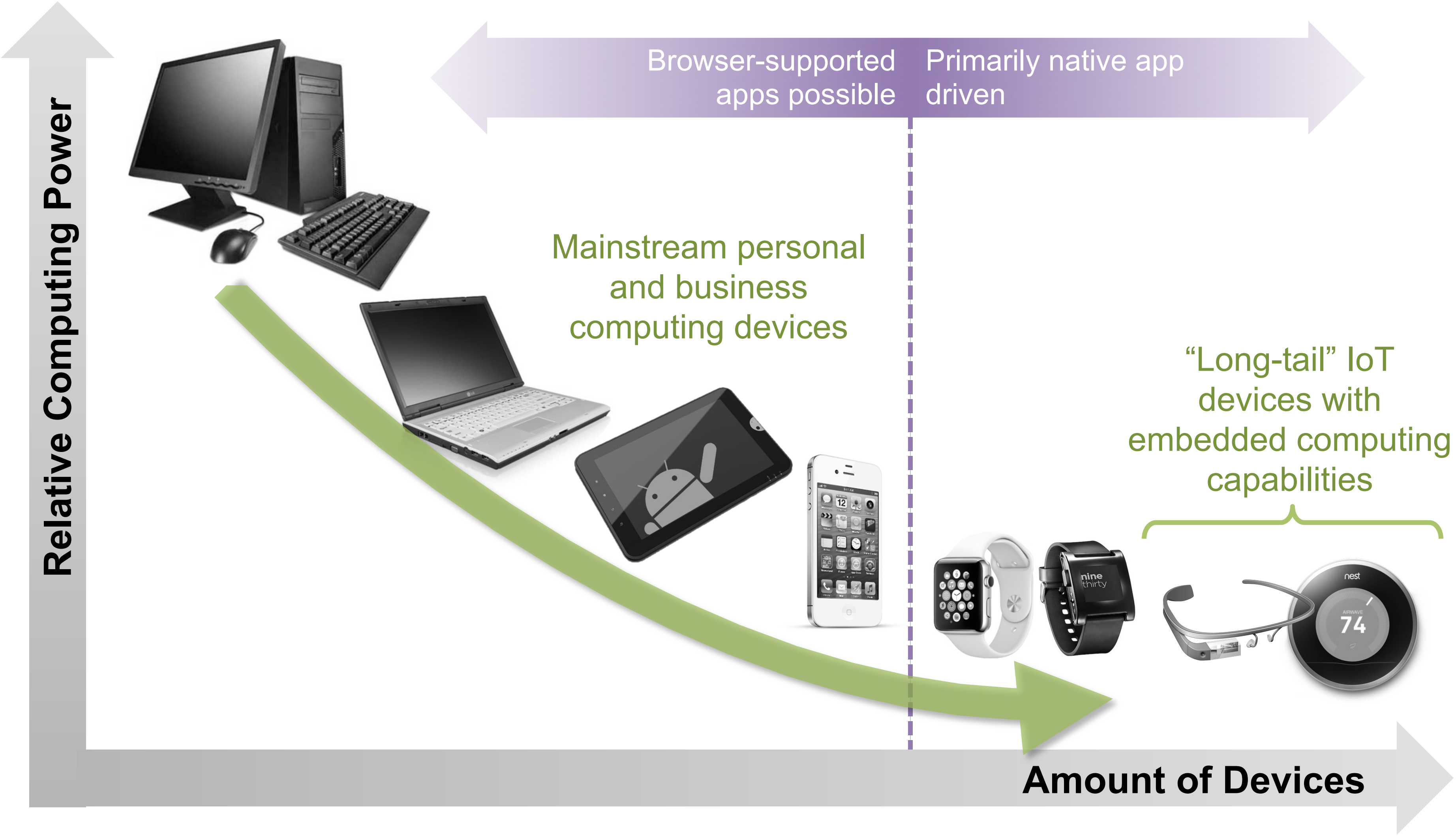
Hackathon?

Ah, so where do hackathons fit in?...

After you've gotten the rest of it right, maybe!

Improving Developer Ecosystem Adoption

Forecast: "Resurgence" of Native Apps, More Fragmentation



Browser-supported apps possible
Primarily native app driven
Mainstream personal and business computing devices
"Long-tail" IoT devices with embedded computing capabilities
Relative Computing Power
Amount of Devices
nine thirty
nest
AIRWAVE
74

Development Expectation Challenges

LESS TIME

Development cycles down to 6-8 months or less, from 12-18

LESS BUDGET

Development budgets not increasing to match mobile needs

MORE COMPLEX

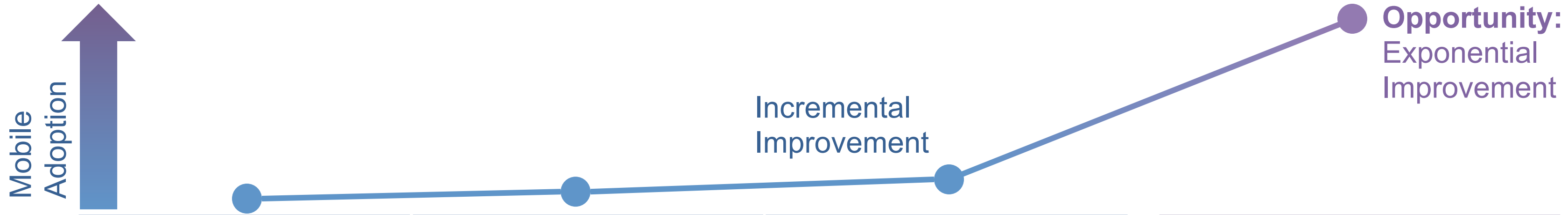
Complexity across **Front-end, back-end** and **secure, scalable integration**

MORE DYNAMIC

Development, Design, Develop, Test all work on the same cadence: **Rapid Prototyping and Ongoing Updates**

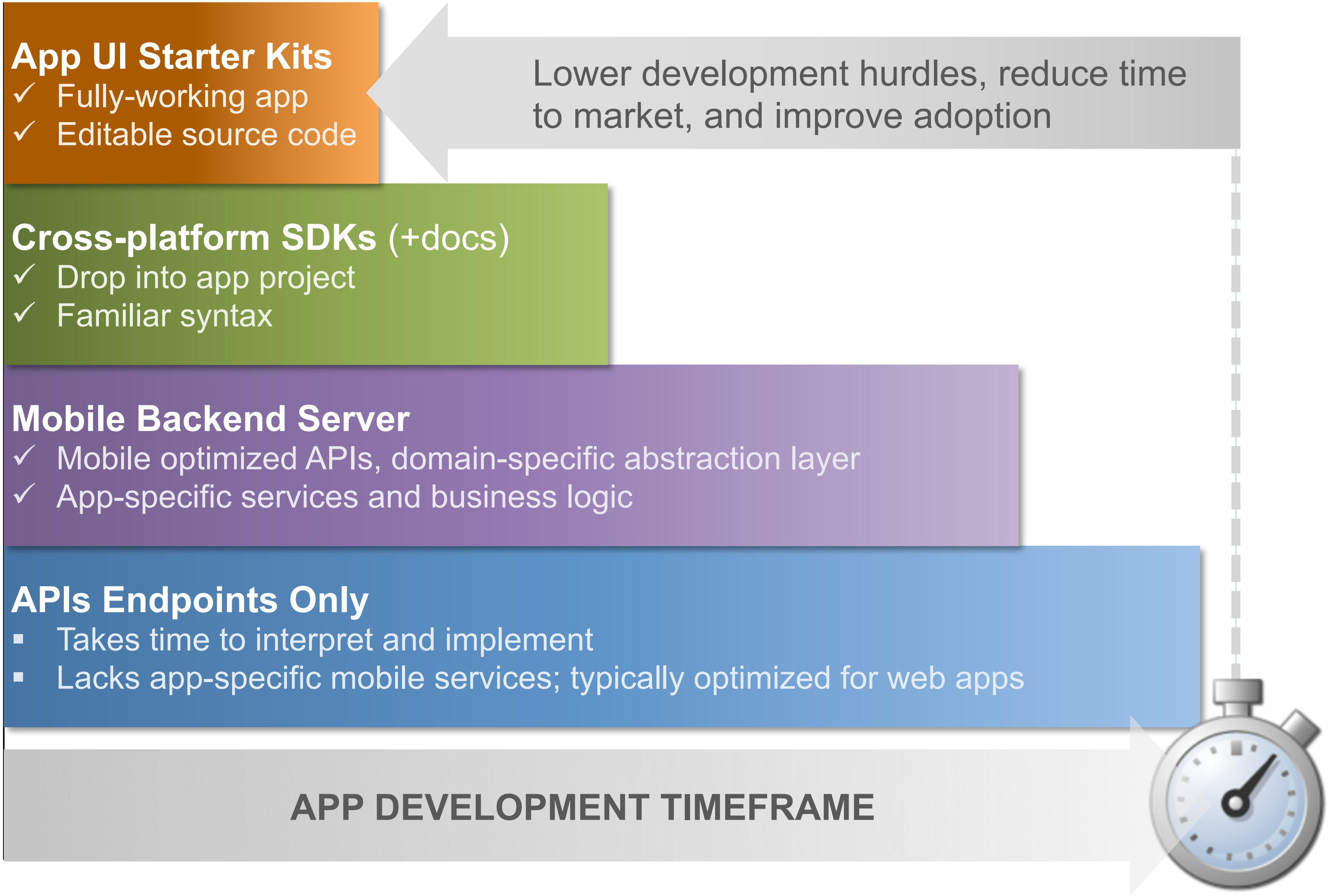
Source: Forrester Research

API Mobile Adoption Challenges and Opportunity



	Documentation	Static Sample Code Snippets	Static Sample App Code	Live Apps, Dynamic Sample Code
Method	▪ Documentation on API methods	▪ Snippets of code in a few languages ▪ API Explorer	▪ Provide a link to some example app in one platform	▪ Live, interactive sample app ▪ Fully working app with source code and SDK
Challenge	▪ Limited ability to understand context of methods ▪ Docs can get out of date	▪ Difficult to maintain ▪ Not customized to a specific use case	▪ More difficult to maintain (requires multiple skillsets) ▪ Not customized to a specific use case ▪ Limited data sets	▪ Maintaining code base across multiple platforms ▪ Providing server-side deployment options for complex scenarios
Result	✘ Poor adoption ✘ Takes longer to build apps	✘ Doesn’t provide a fully working example ✘ Takes longer to build apps	✘ Expensive to build and maintain ✘ Addresses limited use cases	✔ Significantly improve developer adoption ✔ Faster time to market with customized sample app and portable source code

Beyond APIs: Improving Ecosystem Adoption



App UI Starter Kits
- ✓ Fully-working app
- ✓ Editable source code

Lower development hurdles, reduce time to market, and improve adoption

Cross-platform SDKs (+docs)
- ✓ Drop into app project
- ✓ Familiar syntax

Mobile Backend Server
- ✓ Mobile optimized APIs, domain-specific abstraction layer
- ✓ App-specific services and business logic

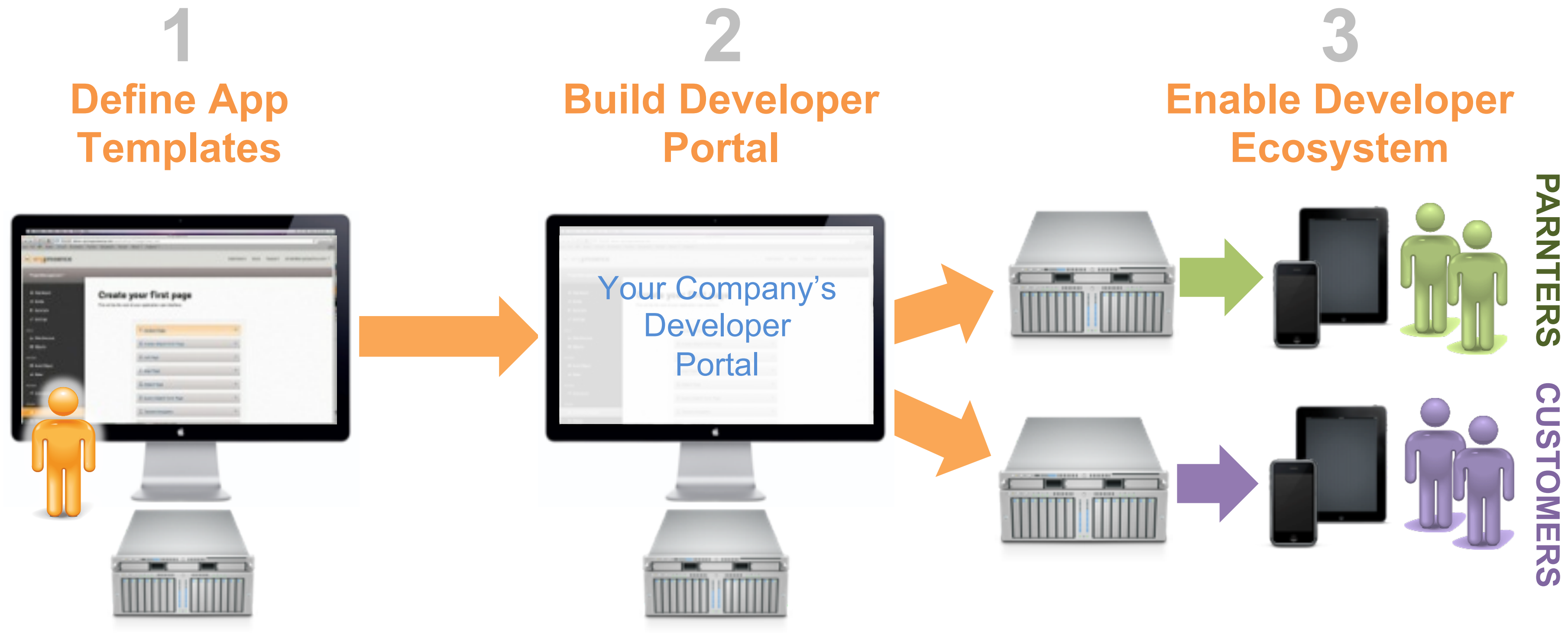
APIs Endpoints Only
- ▪ Takes time to interpret and implement
- ▪ Lacks app-specific mobile services; typically optimized for web apps

APP DEVELOPMENT TIMEFRAME

Brief Demo:

Enhancing Your Developer Portal

How It Works



- Define templates with pre-built integration to your product, service, or content
- Determine which features can be customized by partner or customer developers

- Custom-branded developer portal or new functionality within existing solution
- Enables users to select a template, specify options, and generate an cross-platform apps

- Download source code for backend server, SDKs, and UI starter kits (option to compile apps and host server)
- Functional app with customizable code and deployment options

Use Cases and Key Benefits

Internal Use

For use within the organization to build and deploy apps faster

Key Benefits:

- ✓ ***Efficiency / Scale***
- ✓ ***Enforce IT Policy***

Partner Use

For use by approved partners to enable specific functions within their apps

Key Benefits:

- ✓ ***Partner Enablement***
- ✓ ***Retain Control***

External Use

For use by external developers, to encourage adoption of public APIs

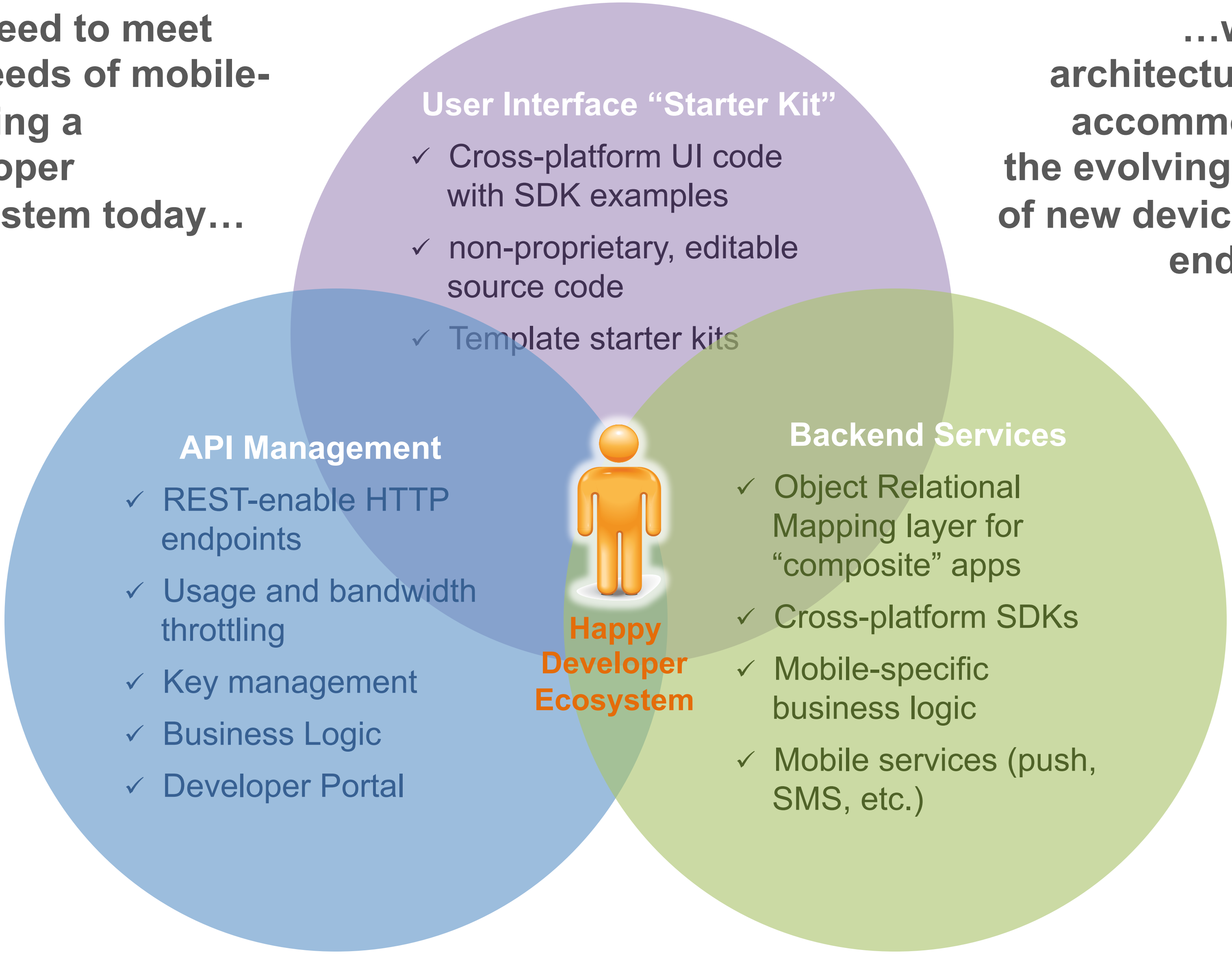
Key Benefits:

- ✓ ***Adoption***
- ✓ ***Revenue / Brand Recognition***

Lessons Learned: **Evolving Developer Ecosystem Needs**

You need to meet the needs of mobile-enabling a developer ecosystem today…

…with an architecture that accommodates the evolving needs of new devices and endpoints



User Interface "Starter Kit"

- ✓ Cross-platform UI code with SDK examples
- ✓ non-proprietary, editable source code
- ✓ Template starter kits

API Management

- ✓ REST-enable HTTP endpoints
- ✓ Usage and bandwidth throttling
- ✓ Key management
- ✓ Business Logic
- ✓ Developer Portal

Backend Services

- ✓ Object Relational Mapping layer for "composite" apps
- ✓ Cross-platform SDKs
- ✓ Mobile-specific business logic
- ✓ Mobile services (push, SMS, etc.)

Happy Developer Ecosystem

Thank You! Questions?

Carlo Longino
VP of Developer Program Services
carlo@wip.org
www.wipfactory.com

Richard Mendis
Chief Product and Marketing Officer
rmendis@anypresence.com
www.anypresence.com