

# Mobile Development @ PayPal

Usman Abbasi

Senior Manager, Mobile Development

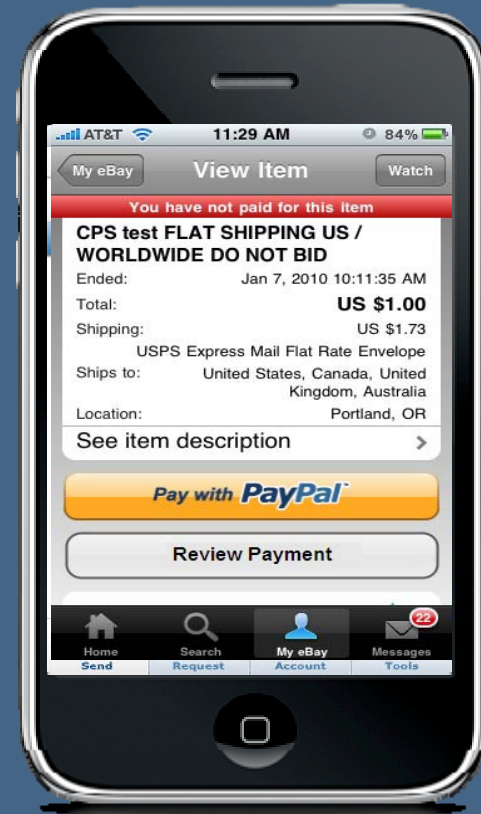
# Backdrop

- Technology is changing fast!
- Lots of competition/innovation
- Fragmentation is continuing problem
- Uncharted territory

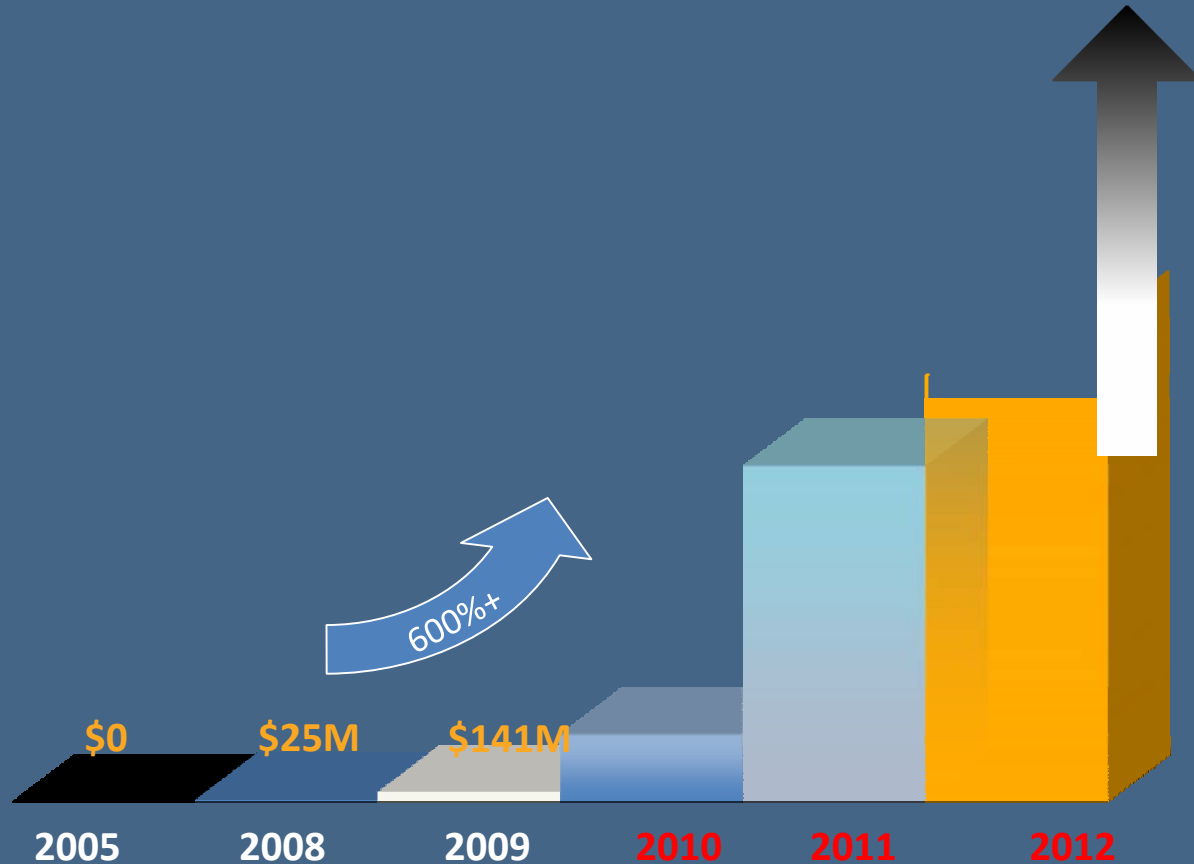


# What is PayPal doing?

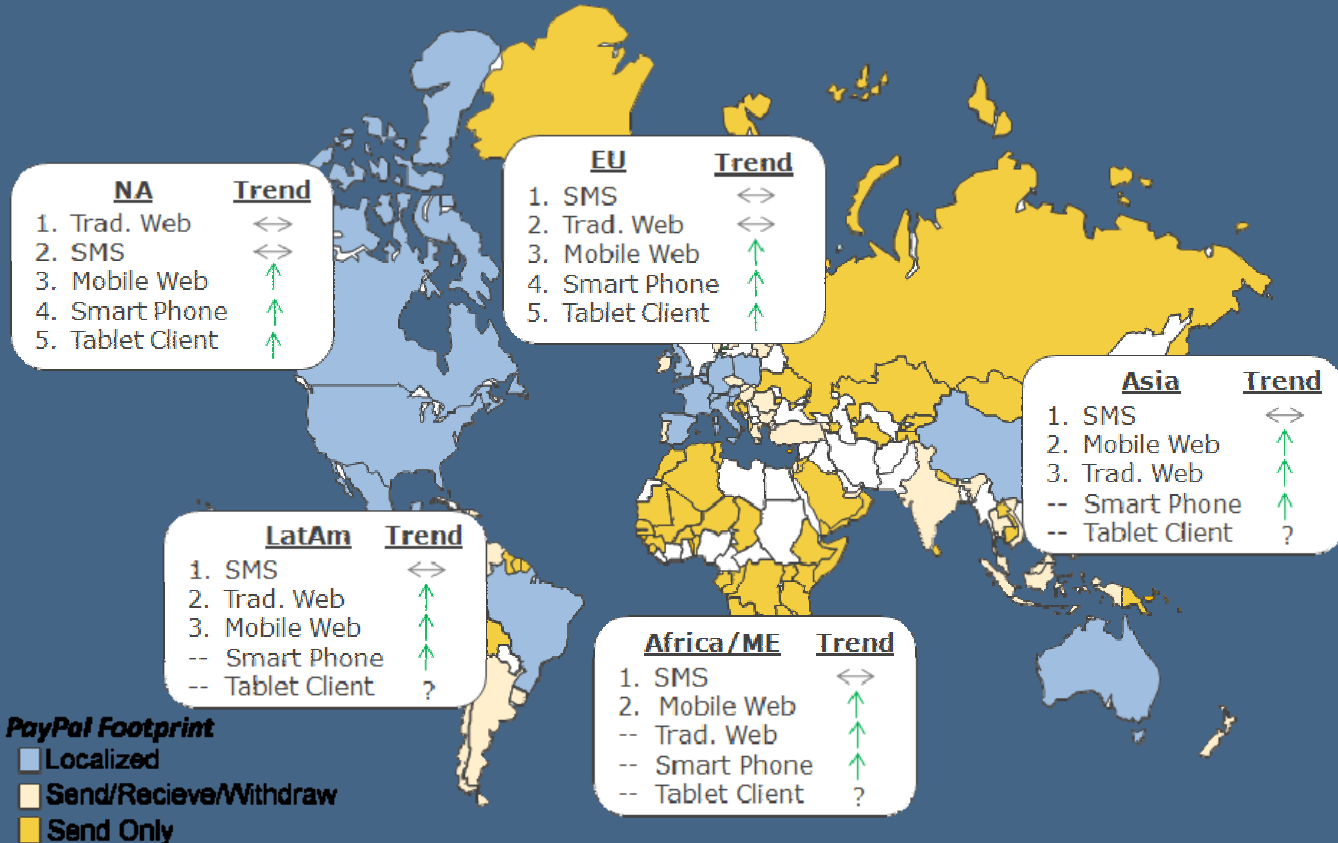
- Consumer Application
  - Access all payment methods anywhere
  - Send/Invoice Money
  - In-Person payments via Bump
- Merchant Checkout
  - In-app, rich authentication
  - 1-click payment, sign-up and sign-in
  - Single integration
  - Guest checkout
  - Auto-detect mobile web



# Huge Growth and Opportunity



# Global Perspective



# Platform Survey



## iPhone

- Flagship platform
- Easier for C/C++ & Mac developers
- Walled garden
  - Good: Protected, uniform platform
  - Bad: TOS causes confusion with every change



## Android

- Flagship platform by 2011
- Easier for Java Developers
- Fragmentation—only going to get worse
- “Open Platform”
  - Good: Flexible and adaptable
  - Bad: Evolving security model

# Portability Frameworks



## Benefits:

- Cost & Time to Market
- Easier to hire—Javascript vs Objective-C
- Easier to pickup
- Good for rapid prototyping

## Challenges:

- Lock-in!!!
- Need native control over UI
- Write once, debug everywhere!
- Obfuscation over long run
- Feature Lag

# Mobile Web

- Mobile Web helps with portability
- Rendering technology helps with fragmentation
- Webkit has paved the way for great experience
- HTML5 brings rich experience



# Bottom Line

- Mobile commerce is already bigger than you think
- Mobile is in flux! New technology announcements daily
- Fragmentation is going to get worse before it gets better. Deal with it!
- Convergence is inevitable but will not be to one platform
- Rules of the road:
  - 1) Global products require a variety of technologies (SMS, Apps, Mobile Web)
  - 2) Technology changes quickly. Expect to change along with it.
  - 3) Prefer Native environments to create rich experiences
  - 4) Do X-platform thru web. HTML 5 is the future
  - 5) Use portability frameworks as prototyping tools