




A CASE STUDY OF INTERNET BUSINESS




Internet Business

- Goal
 - Analyse the degree to which internet business has matured and/or is constantly developing and innovating using a detailed study of an internet business
- 




Internet Business

- Continually developing and growing
 - The main source of growth and development is innovation
 - PayPal is one example of a company focussed on keeping up with development and growth via innovation
- 

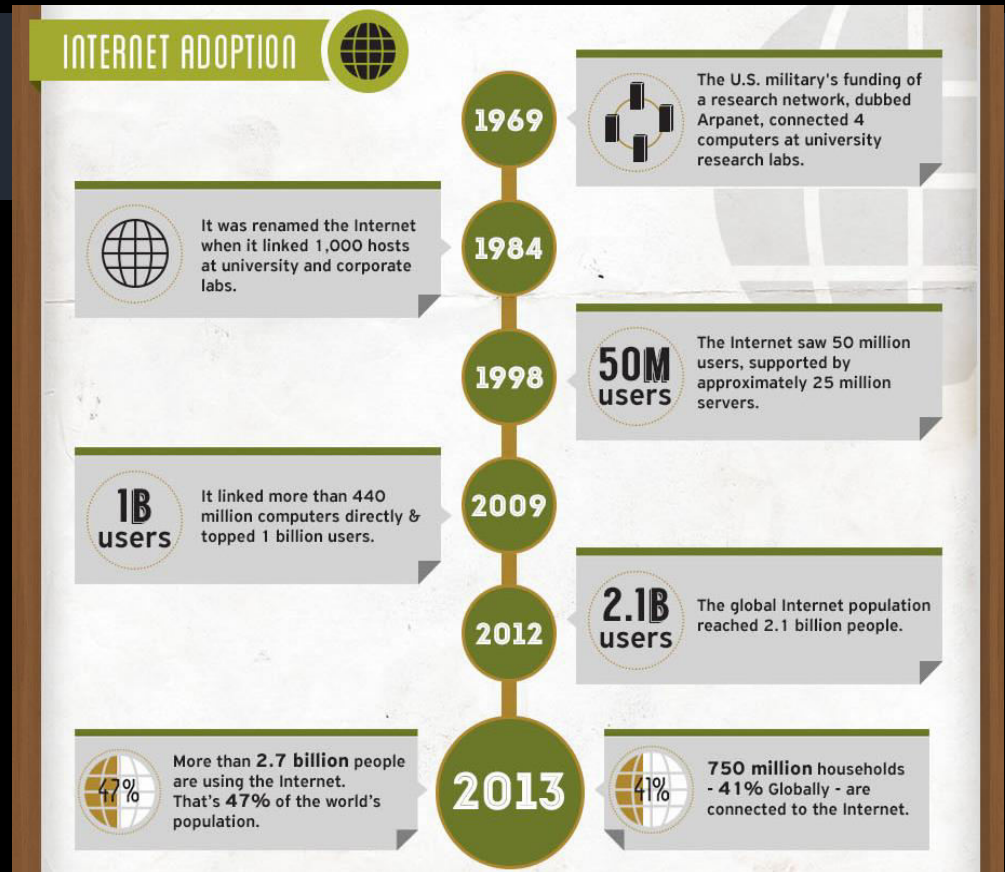


Outline

- Introduce the Web & E-commerce
 - Provide E-commerce development overview
 - Understand the development of the Internet through a detailed case study of PayPal
 - Revolution of PayPal
 - Current developments & innovations
 - PayPal Competitors
- 

The INTERNET

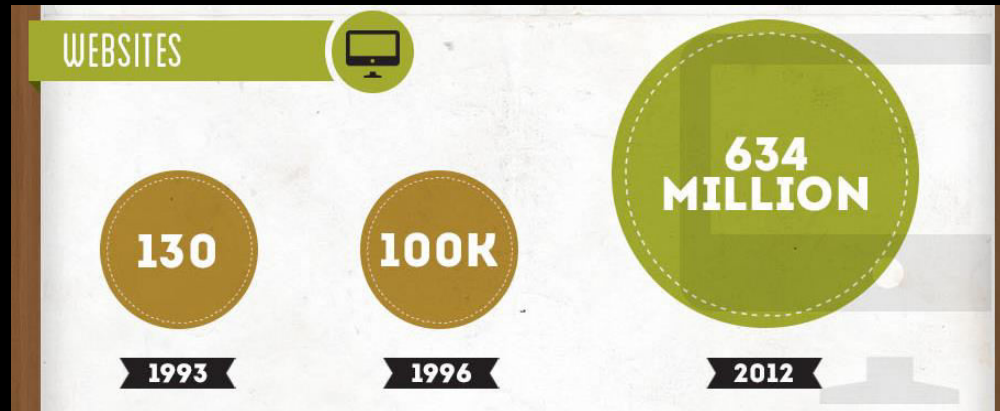
- The internet is “a computer network comprising or connecting a number of smaller networks, such as two or more local area networks connected by a shared communications protocol”
- The internet has been a key resource in allowing the evolution of business where businesses have shifted from real time markets to digital markets, or more commonly known as E-Commerce



(Social News Daily, 2015)

WWW

- The World Wide Web (WWW) was created by Tim Berners-Lee in 1990



E-Commerce

- A type of business model, or segment of a larger business model, that enables a firm or individual to conduct business over an electronic network, typically the internet
- Operates in all four of the major market segments:
 - business to business,
 - business to consumer,
 - consumer to consumer and
 - consumer to business

Percentage Of People Who Have Shopped Online By Area

Half of North Americans said they most frequently purchase from online-only stores, while one-third of Latin Americans prefer sites that also have traditional offline stores. Almost half (47%) of online consumers in the Middle East, Africa and Pakistan say they have never shopped online.





Potential change

- E-commerce *has* changed
 - Accessibility; Broken down physical barriers
 - Security; political and legal legislation
 - Innovation; New business opportunities
 - Communication/Timeliness; Instant global communication

Case study



Establishment



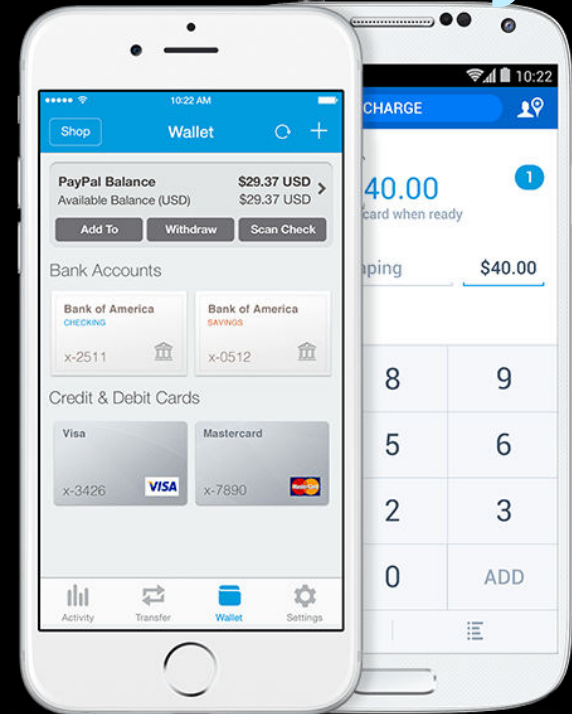
- Founded in 1998 as Confinity, the company merged with Elon Musk's X.com to later be known as PayPal in 2001
- Located in San Jose, California
- Operating with:
 - 26 currencies

PayPal revolutionised Internet business




Accessibility & Flexibility

- PayPal establishes a relationship with both buyer and seller
- PayPal enabled peer-to-peer online transactions by opening its platform, codes and infrastructure to other services
- One of the first payments system to launch an app for iOS and Android






Safety & Security

- Introduction of Secure Sockets Layer (SSL) technology for data encryption
 - PayPal's Seller and Buyer Protection Policy
 - Extra online security through use of self-developed and patent pending method of verification
- 




Payments Integration

- PayPal offers consumers the means to incorporate PayPal functionality into their applications with their application programming interfaces (API)
 - The advantages are :
 - Process one-time and recurring payments without the need to invest in expensive security infrastructures or expensive audits
 - Maintain card and bank account payment schedules without the liability of warehousing payment data
 - Safeguard against un-authorized system access through multi-factor user authentication
 - Centralize payment related activities through Payment Vision administrative site
 - Prevent fraud and returns through ValidPay
- 

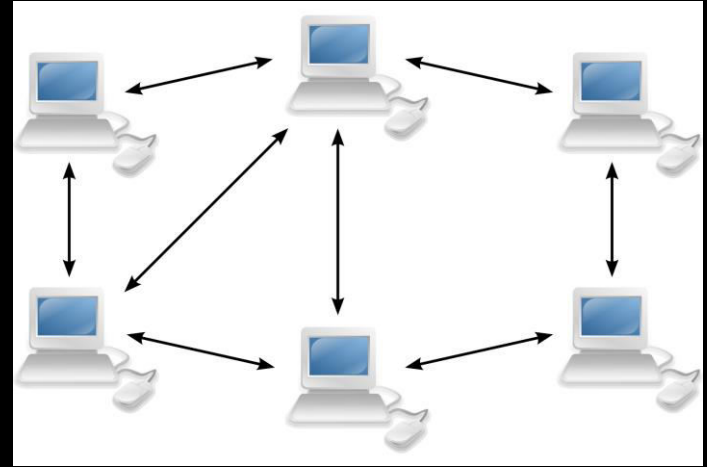


Paypal and innovation

- Taking something existing and bringing it to the masses
 - Taking something existing and fixing problems with it
 - Taking something old and reinventing it
- 

Paypal.me

- Peer to Peer (p2p) payments (cheque, cash, money order, digital transfer) using a unique URL
- P2P is not a new concept, and using it to send payment isn't either – it's not even new for PayPal
- Capitalising on 'everyone' having internet access
- Taking something existing and 'fixing' it – making it simpler



Innovation through partnerships

- PayPal drive a lot of their growth and ability to innovate through strategic partnerships, mergers and acquisitions
 - eBay – PayPal as their growth driver
 - Discover Financial Services (US) → Payment cards
 - POS systems (Home Depot, Abercrombie & Fitch)
 - Third Party developers
 - Improved mobile app
- } Offline Focus

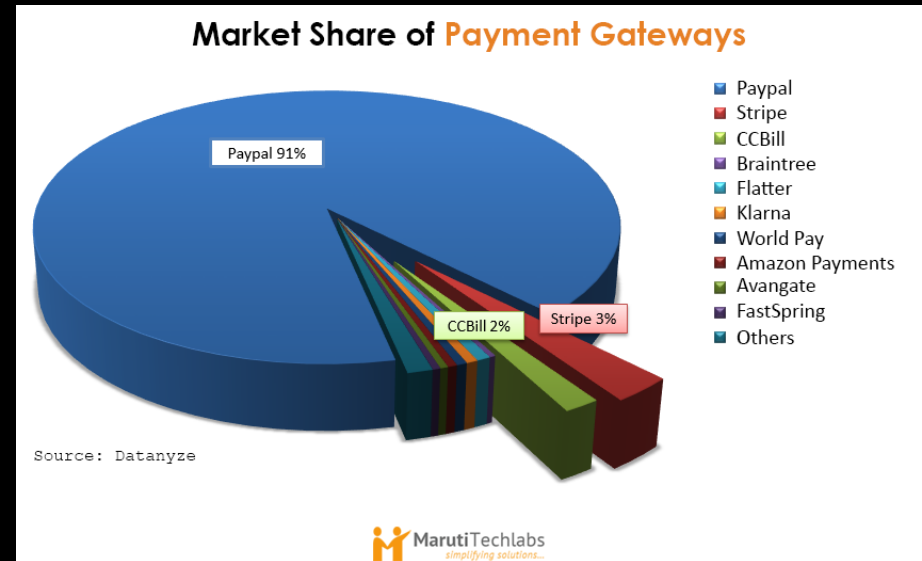


Current standing

- Innovation in the sector is speeding up and PayPal faces new competitors and a need to stay at the edge of new hardware and software developments
- Growth Strategy
 - Centre of next innovation
 - Digital and physical commerce
 - Digital wallet → Mobile wallet → No wallet
 - Focus on the ability for consumers to make payments without relying on cash *or* cards
 - The focus is on the consumers, not the way you pay
 - Re-imagined consumer shopping experience
 - Changed consumer shopping experience
 - “Differentiated and frictionless payment experience across all channels, devices, and all methods of payment

Other Payment Gateways

- Market share – 91%
- Apple Pay?
 - Within the Apple system
- Stripe
 - Integrated by 3,558 new websites in 2015
 - PayPal lost 2,150 websites in the same period
 - Improved technology?



High Scalability of Paypal

- Paypal take a billion hits a day system
 - That might traditionally run on a 100s of VMs
 - But, Paypal shrink it down to run on 8 VMs
 - Stay responsive even at 90% CPU at transaction densities Paypal has never seen before
 - With jobs that take 1/10th the time, while reducing costs and allowing for much better organizational growth without growing the compute infrastructure accordingly


PayPal moved to **Akka + squbs**

High Scalability of Paypal

- PayPal, a system with the characteristics:
 - Scalable, both horizontally to hundreds of nodes and vertically to very many processors, handling billions of requests per day
 - Low latency, controllable at a very fine grain
 - Resilient to failure
 - Flexibility in adjusting the service boundaries
 - A programming model AND culture encouraging scalability and simplicity, including clean failure and error handling



High Scalability of Paypal

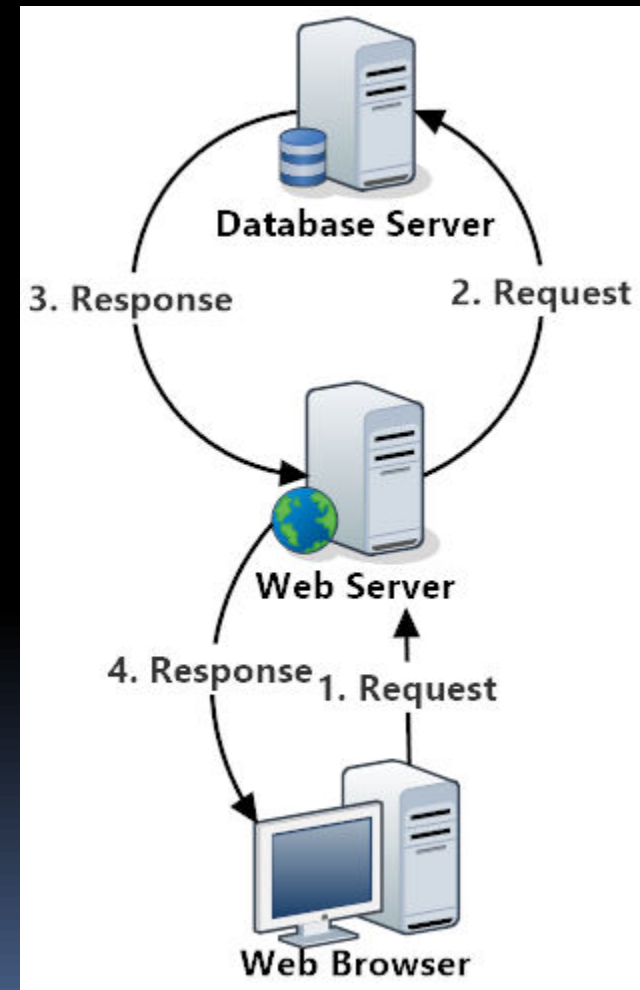
- PayPal wanted to achieve, by:
 - Write code that's easy to test
 - Handle errors and failure scenarios more naturally when compared to the traditional model used on the JVM
 - Write faster, resilient, and simpler code with streamlined error handling and fewer bugs
- 

Paypal - Akka

- Akka is a toolkit and runtime for building highly concurrent, distributed, and resilient message-driven applications on the JVM
- Build applications **more easily**
- Actor based model
- Actors can be stateful
 - No longer have to factor round-trip times to SQL Server, Redis, Cassandra, or whatever into the design of applications
 - The application *already has the state it needs to do its job* by the time a request arrives

Akka-No More Traditional Stateless Design

- The traditional way of developing web applications is stateless - and that's a natural consequence of HTTP, itself an inherently stateless protocol
- But, nearly all of the web applications we are responsible for as developers are stateful



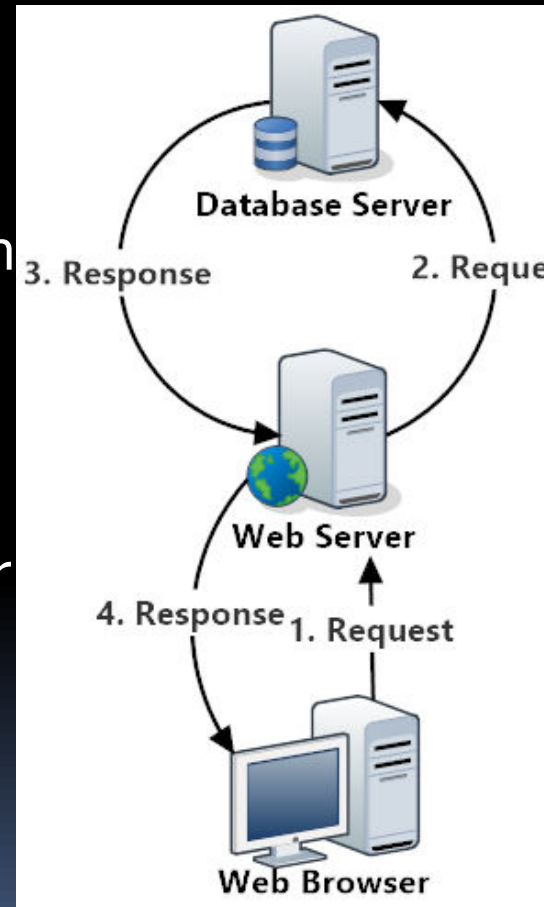
Stateful Applications

Application Type	Stateful Entities
eCommerce	inventory, pricing, availability
Social Apps	updates, graph
Analytics	reports / indicators
Marketing Automation	activity / rules
IoT	device status
Games	multiplayer
Chat / Messaging	messages

Large part of Web Developer jobs consists of managing, mutating, and retrieving state

Stateless Design


- “web application” is really just a stateless command processor
 - All of the information about a user’s session such as their identity and authorization claims, are really stored and retrieved from the database
 - All of the content needed to serve, whether it’s listing products for an eCommerce system or any of the other examples
 - All things - fundamentally live in the database





Stateless Design

- Pros

- It cleanly separates application from the data
 - Easily throw more web servers under a load balancer if traffic increases
 - Easily update of stateless web applications dozens of times a day without worrying about the availability of your data
- 

Stateless Design

- Cons

- **Can't support increasingly popular categories of applications demanded by users**
 - Any application that needs to perform real-time work could never be built using stateless because a need of state locality in order to achieve those response times
 - A multiplayer video game or chat application implemented using a stateless application model
- **Can't support increasingly large workloads**
 - As the Internet is a much larger place now than it was a few years ago and developers are expected to retain and use as much data as possible in order **to satisfy the increasingly high standards of customers**
 - As a result of that, **throwing a lot more information at data tier** than we used to - like multiple orders of magnitude more data
 - And those databases can't scale to support arbitrarily large workloads, especially write-heavy ones
 - Needed to rethink the way to store and retrieve data in order to support the larger units of work while still maintaining fast response times

Stateless Design-Solutions

- Many architects are being forced to reconsider the decades-old stateless “get the data into SQL Server as fast as possible” approach to web application design
 - One solution many architects are considering for solving the “data layer scalability problem” is NoSQL
 - This approach certainly worked (Cassandra and Hadoop,) but it didn’t workout for a soft real-time requirement that is essential for the success of the business
 - Second solution
 - Statefull Design of Application

Stateful Design

- Idea
 - Application is the single source of truth for the “state” of things, and your database is just cold storage / backup
 - The way to architect a typical web application to be “stateful” :



Stateful Design

- Retain the stateless Web UI / presentation layer
 - To keep the benefit of being able to deploy changes to UI with zero risk to the integrity of state → Achieved by this design
 - But add a new, “middle” application tier that sits between the Web UI and the database → Novelty to this design




Stateful Design

- The stateful middle service is responsible for doing the following:
 1. **Accumulating state**, in-memory, and asynchronously flushing updates in critical state to a durable store
 2. **Re-hydrating its state**, on-demand, from the durable store
 3. **Fulfilling requests from other services** using its state
 4. **Reacting to changes in state as they occur in real-time** (Most important)



Stateful Design

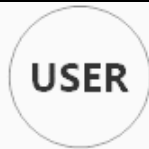
- Instead of polling the database for answers continuously, which increases contention, I/O, response times, and error rates
 - Simply react to changes interested in as they occur
 - This change can produce a multiple order of magnitude increase in throughput, responsiveness, resiliency, and scalability of the system
- 



Stateful Design

- By treating the database like merely a durable store for looking things up and not the beating heart of application, many of the problems we run into with the traditional model will go
 - For instance, suppose building a marketing automaton system and customer wanted to send one of their users a message after that user had triggered events 0-3, often in *rapid succession*

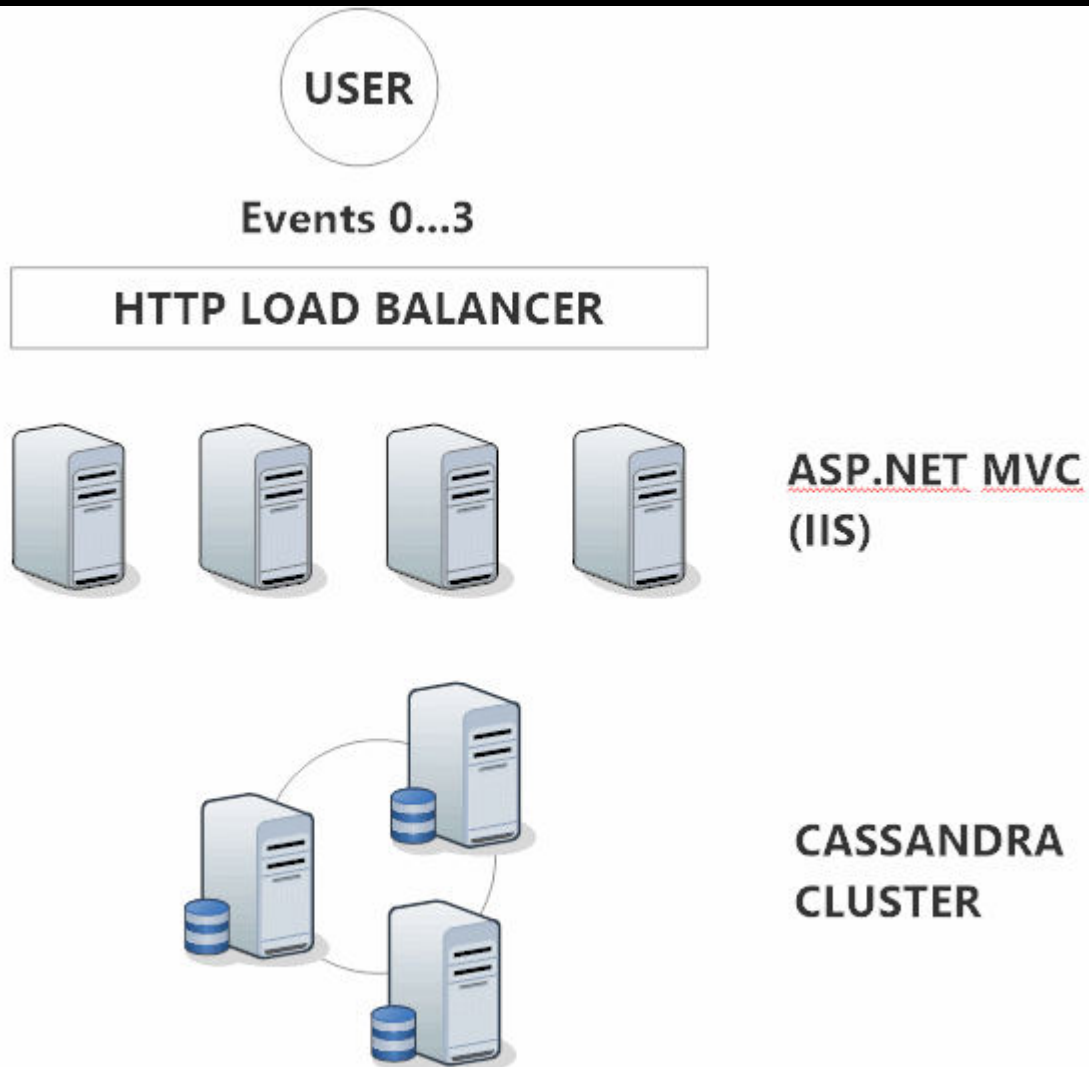
StateLess Design - Theory



If the database to handle everything then

- Stateless web application would write the event currently being observed to the database, and then read that event **plus** all of the other accumulated events out of the database
- Eventually, when the last event was observed, send the message to the end-user because the database would have returned all four events

StateLess Design - Reality




Idea fails miserably
- even under the
most idealized test
environment

1. Consistency of the data store at the time of query, even under ACID
2. Concurrency of the events and requests happening at the level above the database



StateLess Design - Reality

- Retrieved Values are chaotic and random
 - Because trusting the database to be responsible for the ordering of the messages , and
 - To have an immediately consistent view of that data whenever require, ignoring any other traffic or network conditions that might be affecting the application as a whole or the database specifically
- 

Stateful-Reality



- Under the stateful programming approach, design system such that all events for an individual user are always routed to the exact same machine and the exact same actor in network
- Uses Consistent Hashing Router concept

Stateful Design-Reality

- The events can arrive in any arbitrary order, interleaved with other events potentially, and multiple web servers operating under the load-balancer can each forward one or more of these events to the stateful application server responsible for accumulating them for this user
- But once all four events are observed, this fact is both consistent and well-known to the application *immediately* without needing to talk to a database
- The result is a predictable product with low overhead, great scalability, and *extremely fast* response times



Stateful Design

- Requires more care and caution to deploy updates to the stateful application and requires the developers to learn how to manage state in a concurrent + distributed fashion
- Design technique to meet the standards of today's users and information economy
 - ▣ Taking control of state back from the database
- Even used by Facebook, Uber, and Microsoft




SQUBS

- A new stack (pronunciation rhyming with “cubes”)
- The standard for building Akka-based reactive applications @PayPal
- Creates a modular layer called “cubes” that are symmetric to other cubes
 - ▣ Unlike libraries with concrete dependencies at the API layers, cubes ride on the actor system and **only expose the messaging interface already provided in Akka**



Conclusion

- Wide range of growth and development
 - Certainly innovating in their field, and expanding beyond it
 - What will be the next approach...
- 

References

- <https://petabridge.com/blog/stateful-web-applications/>
- Auchard, E. (2007). PayPal offers secure way to shop non-PayPal sites. Reuters. Retrieved from <http://www.reuters.com/article/2007/11/19/us-ebay-paypal-ecommerce-idUSN1840572920071119>
- Australian Treasury. (2012). Australian Small Business Key Statistics and Analysis: Chapter 11. http://www.google.com.au/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=4&ved=oCDkQFjADahUKEwiAkbe5_4TlAhVHGpQKHedUCds&url=http%3A%2F%2Fwww.treasury.gov.au%2F~%2Fmedia%2FTreasury%2FPublications%2520and%2520Media%2FPublications%2F2012%2FAustralian%2520Small%2520Business%2520-%2520Key%2520Statistics%2520and%2520Analysis%2Fdownloads%2FRTF%2FAustralianSmallBusinessKeyStatisticsAndAnalysis.ashx&usq=AFOjCNGnnjbRCl6g6sgASFNKXTz9GjnXrA&sig2=_kzoqAH2tFKDOzmaLQbvaA&bvm=bv.103073922,d.dGo
- Berners-Lee, Tim, Weaving the Web: The Past, Present and Future of the World Wide Web by its Inventor (London, 1999)
- Cheng, R. (2014). eBay, PayPal to split into separate companies in 2015 - CNET. CNET. Retrieved 28 September 2015, from <http://www.cnet.com/news/ebay-to-split-paypal-into-separate-company-in-2015/>
- Chu, L. (2008). What PayPal does with your money. Money.cnn.com. Retrieved 28 September 2015, from http://money.cnn.com/2008/02/26/smbusiness/paypal_float.fsb/
- Cohan, P. S. (2013, September 5). *What PayPal's Rocky Beginnings Can Teach You About Startup Success*. Retrieved from Entrepreneur.com: <http://www.entrepreneur.com/article/228206>
- E-Consultancy. (2011). 10 amazing e-commerce infographics <https://econsultancy.com/blog/7810-e-commerce-infographics/>
- Forrest, C. (2015). How the 'PayPal Mafia' redefined success in Silicon Valley. TechRepublic. Retrieved from <http://www.techrepublic.com/article/how-the-paypal-mafia-redefined-success-in-silicon-valley/>
- Gordon, L. (2014, August 21). *The Four Aspects of PayPal's Current Growth Strategy*. Retrieved from Euromonitor International: <http://blog.euromonitor.com/2014/08/the-four-aspects-of-paypals-current-growth-strategy.html>

- Ha, P. (2008). eBay Acquires Fraud Sciences For \$169 Million. TechCrunch. Retrieved 28 September 2015, from <http://techcrunch.com/2008/01/28/ebay-acquires-fraud-sciences-for-169-million/>
- Heggstuen, J. (2015, June 16). *THE PEER-TO-PEER PAYMENTS EXPLAINER: Why tech companies and banks are offering a service that doesn't make money*. Retrieved from Business Insider Australia: 2015
- History of PayPal Inc. (2004). Nternational Directory Of Company Histories, 58. Retrieved from <http://www.fundinguniverse.com/company-histories/paypal-inc-history/>
- Hoffman, M. (2008). EBay Buys Bill Me Later for Nearly \$1 Billion. Inc.com. Retrieved 28 September 2015, from http://www.inc.com/mikehofman/2008/10/ebay_buys_bill_me_later_for_ne.html
- Investopedia. (2015). Electronic Commerce – ecommerce. <http://www.investopedia.com/terms/e/ecommerce.asp>
- Kane, M. (2002). eBay picks up PayPal for \$1.5 billion - CNET. CNET. Retrieved 27 September 2015, from <http://www.cnet.com/news/ebay-picks-up-paypal-for-1-5-billion/>
- Kerner, S. (2005). EBay Buys VeriSign's Payment Gateway Unit - InternetNews.. Internetnews.com. Retrieved 27 September 2015, from <http://www.internetnews.com/bus- news/article.php/3555201>
- Krishnan, R., Smith, M. D., & Telang, R. (2002). The Economics of Peer to Peer Networks. *H. John Heinz III School of Public Policy and Management*.
- Ludwig, S. (2013). PayPal buys young startup Iron Pearl to help it acquire more users. VentureBeat. Retrieved 28 September 2015, from <http://venturebeat.com/2013/04/11/paypal-buys-iron-pearl/>
- Macsai, D. (2011, March 14). *THE 10 MOST INNOVATIVE COMPANIES IN FINANCE*. Retrieved from Fast Company: <http://www.fastcompany.com/1738549/10-most-innovative-companies-finance>
- Magdurulan, D. (2015). Peer to Peer Networking Diagram. Retrieved from <https://andrewmagdurulan.wordpress.com/networking/>
- MarketLine,. (2013). PayPal: Online payments tycoon (pp. 1-16). London: MarketLine.
- Mishkin, S. (2014, September 30). *Innovation key to PayPal's successful independence*. Retrieved from Financial Times: <http://www.ft.com/intl/cms/s/0/307d8e56-48b7-11e4-9d04-00144feab7de.html#axzz3nkcBw5uv>

- Nickson, C. (2012, April 17). The internet as a global marketplace. Retrieved from <http://www.atechnologysociety.co.uk/internet-global-marketplace.html>
- Oxford University Press. (2015). "Internet, n.". OED Online. <http://www.oed.com/view/Entry/248411>
- PayPalAustralia. (2012, August 14). *Co-op enlists PayPal to help Australian students beat the queues this semester*. Retrieved from PayPal: <http://webcache.googleusercontent.com/search?q=cache:zv9SPooWLi4J:https://www.paypal-media.com/au/press-releases/co-op-enlists-paypal-to-help-australian-+&cd=5&hl=en&ct=clnk&gl=au>
- PayPal. (2015). Logo. Retrieved from https://www.paypalobjects.com/webstatic/mktg/logo-center/PP_Acceptance_Marks_for_LogoCenter_266x142.png
- Payment Online Corporation,. (2009). Payment Gateway software. Payment Online Corporation. Retrieved 28 September 2015, from <http://www.paymentonline.com>
- PayPal Credit,. (2015). PayPal Credit. [Paypalcredit.com](https://www.paypalcredit.com/index.html). Retrieved 28 September 2015, from <https://www.paypalcredit.com/index.html>
- PayPal Developer,. (2015). Developer.paypal.com. Retrieved 28 September 2015, from <https://developer.paypal.com>
- PayPal Inc.,. (2015). Home. PayPal About. Retrieved 27 September 2015, from <https://about.paypal-corp.com>
- Perez, S. (2015, September 1). *PayPal Launches PayPal.Me, A Simpler Way To Request Money Using Your Own Personalized URL*. Retrieved from TechCrunch: <http://techcrunch.com/2015/09/01/paypal-launches-paypal-me-a-simpler-way-to-request-money-using-your-own-personalized-url/#.zb4yal:vMrN>
- Rao, L. (2015, September 1). *PayPal makes money transfers more personal*. Retrieved from Fortune: <http://fortune.com/2015/09/01/paypal-me-payments-mobile/>
- Rao, L., Perez, S., & Lunden, I. (2013). eBay's PayPal Acquires Payments Gateway Braintree For \$800M In Cash. TechCrunch. Retrieved 28 September 2015, from <http://techcrunch.com/2013/09/26/paypal-acquires-payments-gateway-braintree-for-800m-in-cash/>
- Regan, K. (2002). PayPal IPO off to spectacular start. E-Commerce Times. Retrieved from <http://www.ecommercetimes.com/story/16368.html>
- Reserve Bank of Australia,. (2011). Strategic Review Innovation. Retrieved 28 September 2015, from <http://www.rba.gov.au/payments-system/reforms/strategic-review-innovation/submissions/201106-strategic-review-innovation/paypal.pdf>
- Rubin, B. F. (2015, July 20). *Now independent, PayPal worth nearly \$50 billion*. Retrieved from Cnet: <http://www.cnet.com/news/now-independent-paypal-worth-nearly-50-billion/>
-

- Rupley, S. (2009). PayPal's (Partially) Open Platform to Usher in New Payment Models & Apps. Gigaom. Retrieved 28 September 2015, from <https://gigaom.com/2009/11/03/paypals-partially-open-platform-to-usher-in-new-payment-models-apps/>
- Ryan, J. (2010). History of the Internet and the Digital Future. London, GBR: Reaktion Books. Retrieved from <http://www.ebrary.com>
- Social News Daily. (2015) The Internet Then And Now [Infographic]. <http://socialnewsdaily.com/15544/the-internet-then-and-now-infographic/>
- Team, T. (2013, March 4). *Here Are Some Factors Driving PayPal's Rich Growth Prospects*. Retrieved from Forbes/Investing: <http://www.forbes.com/sites/greatspeculations/2013/03/04/here-are-some-factors-driving-paypals-rich-growth-prospects/>
- Techlabs, M. (2015, May 1). *Why Stripe can emerge as the Market Leader in Payment Gateways*. Retrieved from Maruti Techlabs: <http://www.marutitech.com/why-stripe-can-emerge-as-market-leader-in-payment-gateways/>
- The history of online shopping in nutshell. (2010, March 26). Retrieved from <http://www.instantshift.com/2010/03/26/the-history-of-online-shopping-in-nutshell/>
- Williams, D. (2007). Pro PayPal E-Commerce. Berkeley, CA: Apress.