PUSH TECHNOLOGY™

Solving Problem Apps

How Realtime Messaging overcomes common problem app issues

Contents

Introduction

Mobile, Web and IoT apps can be thought of in one of two ways – dream apps and problem apps. Dream apps are easy to name – WhatsApp with 900 million users, KakaoTalk with 170 million users, Instagram with 400 million users or Snapchat with 6 billion daily video views. The clue is in the numbers – there are millions of users. This is the aspiration of all apps developed today – achieve great app adoption by delivering a great experience.

Problem apps plague app stores, developers, startups and many established global enterprise organizations – achieving notoriety for all the wrong reasons.

And this is where we face problem apps. Problem apps plague app stores, developers, startups and many established global enterprise organizations – achieving notoriety for all the wrong reasons. Common complaints from users – the app is too slow, the experience is inconsistent, and there is no interaction between the user and the app or a lack of personalization.

Any person responsible for app development and adoption needs to overcome these common problems. This whitepaper discusses why effective data integration and delivery is core to successful apps, the common problems developers face, and the steps to overcome them.

Problem Apps Defined

A vast majority of app users will only give a problem app three chances.

Users are intolerant of problems with an app, especially repeated problems, and are quick to abandon according to a mobile app survey by Dimensional Research. A vast majority of app users will only give a problem app three chances. 80% indicated they will only attempt to use a problematic app three times or less, 53% uninstalled or removed a mobile app with severe issues like crashes, freezes or errors and 36% will stop using a mobile app due to heavy battery usage (usually a sign of poor data handling).

Furthermore, the same research showed that users blame the mobile app and the company who made it - more than half (55%) see a company's brand devalued when a mobile app crashes or produces errors.

What Makes a Problem App?

1. **App doesn't deliver a realtime experience** – A new performance benchmark from TechBeacon highlighted that 61% of users want apps to start in four seconds or less and 49% expect apps to respond in two seconds a less. 7% want a mobile app to respond in less than one second. If you have 100,000 app users, that's 7,000 users you could lose for not being realtime.

   Many of today's mobile applications are static representations of data, refreshing the information only when a user explicitly asks for an update. But interactive applications are infinitely more engaging, updating in realtime as new data becomes available, without the user having to ask. To deliver realtime experiences though, there are more problems:

- Web (RESTful) APIs have become the de facto standard for mobile app integration, but this requires the applications to repeatedly poll for updates, doesn't support reconnection well, and doesn't offer a realtime user experience.
- Mobile app users are frustrated by poor performance and reliability – often caused by the unknowns of delivering data via the Internet and mobile networks.
- Interactive applications demand bi-directional communication for a truly collaborative experience across multiple users.

2. **App cannot scale.** Headline grabbing app crashes such as 'Starbucks Users Steam over "Venti" iPhone App Snafus', 'Truly awful: Centrelink, Medicare apps slammed' or 'iPhone 4S Release Crashes App Web Site' are bad news. In such cases, while apps are developed hoping for success, investment upfront isn't done because success is uncertain, it's complex, time-consuming and could be money down the drain. 37 % of users will think less of a brand if its mobile app crashes or causes errors. Nearly half of users will remove the app, damaging your revenue stream.

   To build scalable apps, microservice architectures have become important. Microservices is an architectural style in which complex applications are composed of small, bounded and loosely-coupled services that can evolve, scale or even disappear with total independence. However, challenges arise:

   - To reduce service coupling and improve scalability, microservices must be able to exchange data objects without being aware of each other.
   - HTTP resource APIs are under-performing, and the request/response pattern inherently causes unwanted coupling.
   - An event-driven integration pattern will achieve the best loosely-coupled architecture, but will introduce data complexity.
   - Legacy middleware and messaging patterns aren't designed for a cloud-native microservices architecture.

3. **App does not support realtime events or live streaming.** It is widely believed that millennials are embracing event-driven and live streaming mobile apps and adopting them at record rates. Gartner says you need to be ready for it especially as the Internet of Things (IoT) enters the mix. Gartner even went so far as to claim that centralized systems will not work and a distributed architecture is required for IoT with data being "fanned-out" closer to the edge. Your apps, especially IoT use cases, will need to rely on event-driven data integration, be ready to handle unexpected traffic and casually connected devices. So consider how your app can recover when the network lets you down, and the data distribution technology you need to handle this.

   When developing apps, the problem is that many integration architectures were never designed to work across the Internet, let alone handle event-driven data.

Your apps, especially IoT use cases, will need to rely on event-driven data integration, be ready to handle unexpected traffic and casually connected devices.

4. **App wastes bandwidth.** The Internet isn't free. Someone, somewhere, whether it is the end user or the business is paying for the use of it. If an app is burning through network resources when transporting data, it is burning through money as well. With apps popping up allowing users to monitor data-slurping offenders, no app can afford to waste data.

5. **App offers no operational savings.** Mobile applications can transform a wide range of business tasks and remove the unnecessary and unpopular technology barriers many users hate. Use cases are widespread, from enabling remote workers, to creating brand new customer support products, or tools that make field operations more efficient. These business apps should deliver real cost savings - not to mention improve employee satisfaction. To succeed, your apps need secure access to all relevant data from a single integration point and should reduce and automate user tasks.

To succeed, your apps need secure access to all relevant data from a single integration point and should reduce and automate user tasks.

6. **App is irrelevant to customers/doesn't meet their needs.** Huffington Post complained in an older article about the Weather Channel App. Why? "Should you bring your umbrella for today's commute? Your Weather Channel App might tell you -- if it feels like updating to today's date. Too often we've left our rainboots by the front door because our app was stuck on yesterday's forecast." This complaint is a few years old now, but still holds true today. Apps – and therefore users – should be able to rely on continuous, intelligent data updates using realtime integration.

7. **App doesn't save time.** We've established consumer impatience. If an app doesn't save time, adds extra steps to simple processes, or is slow, it won't be adopted and you have a problem. An easy example is when customer information stored in filing cabinets takes too much valuable time to locate. Apps should save time by making processes easier – to do this a reactive data layer that rationalizes data resources into a single flexible model allows developers to build features rather than integration.

8. **App takes too long to develop, wasting resources.** One app developer on Quora just recently asked 'Am I taking too long to develop my app?' The first response from an Android developer:

   "Building apps takes a little time. Building good apps takes a lot of time. Building a perfect app takes a hell lot of time and if you are the only one contributing to the project, brother don't even expect you are going to do it in a few days. I have been developing an app since 5 months and I haven't even completed half of it… My advice to get it quicker- you build the core framework and design and ask your friends to do the easier and boring parts."

   My advice to get it quicker- you build the core framework and design and ask your friends to do the easier and boring parts.

   Another question on Quora 'How long does it take to build a mobile application?' Answers included:

   "One of the primary determinants of how long it takes to build a mobile app is the number of features involved... As you can imagine, the more features an app has, the more time it takes to design, develop, and test. We encourage clients to focus on building *Minimal Viable Products* - the minimum amount of functionality required to provide value to your customers. Building an MVP keeps the overall number of features focused, creates the first version of your mobile app sooner, and allows additional features to be developed through subsequent releases as you learn more about your user's needs and feedback."

   "At our company, most of the projects take around 2-3 months to go from idea to app store. This of course depends on what you want to build, but for an MVP of an idea, you generally don't want your development time to exceed past this."

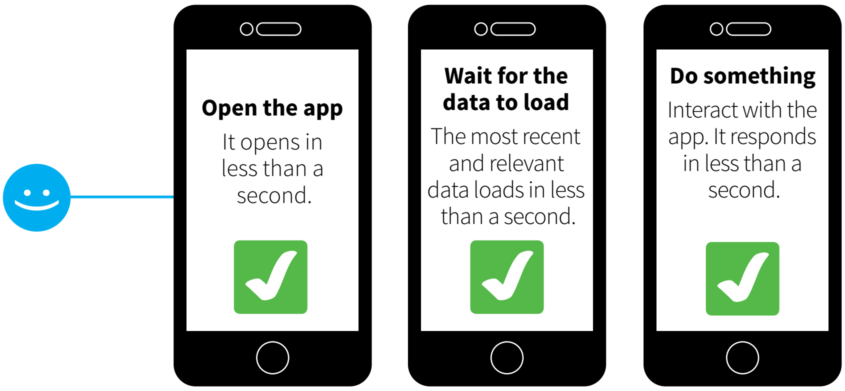
   The point here is that app development takes time. Short term business objectives and time-to-market demands often mean that tactical integration decisions are made for many cloud and mobile applications, rather than considering a scalable long term view across application architecture. MVP gets you to market, but often these apps cannot be onward developed, scaled, maintained, or even retired!

   Short term business objectives and time-to-market demands often mean that tactical integration decisions are made for many cloud and mobile applications, rather than considering a scalable long term view across application architecture.

9. **Apps don't reuse components.** This is all about budget. With many separate teams, each with their own budget to launch a mobile product, there is no incentive for any one team to invest in reusable integration. This isn't scalable, wastes development resources, and leads to all manner of issues around data integrity and security. IT must step up and regain control, mandating standardized integration and enabling unified visibility across data distribution.

Data is Core to Successful Apps

Stop for a minute and think about your favorite app – a social, travel, navigation, weather or gaming app. You need to:

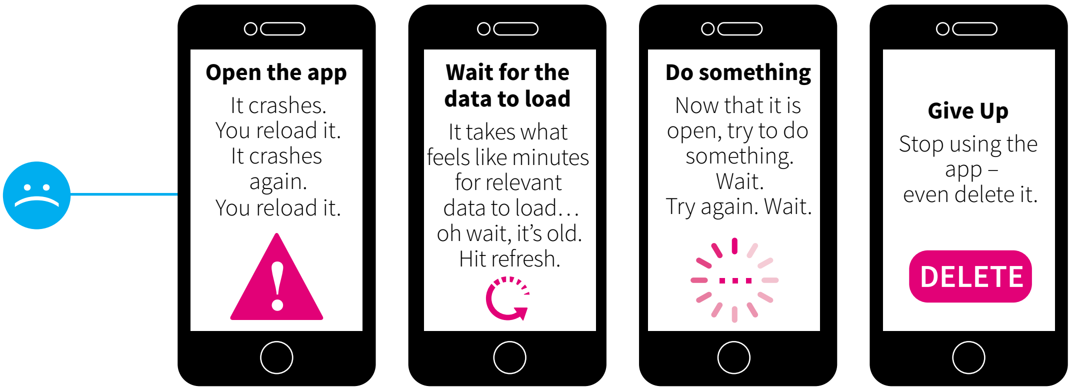


Open the app
It opens in less than a second.
✓

Wait for the data to load
The most recent and relevant data loads in less than a second.
✓

Do something
Interact with the app. It responds in less than a second.
✓

Now think about an app you hate. Common issues include the app crashing when loading, data that doesn't populate quickly, or users having to wait when they ask the app to do something. App adoption in these cases is probably low.



Open the app
It crashes.
You reload it.
It crashes again.
You reload it.

Wait for the data to load
It takes what feels like minutes for relevant data to load… oh wait, it's old.
Hit refresh.

Do something
Now that it is open, try to do something.
Wait.
Try again. Wait.

Give Up
Stop using the app – even delete it.
DELETE

The best apps are the ones that have the data users want, when they want it. However, integrating apps into established backend systems, complex databases, or legacy middleware is hard – made harder as data and services now sprawl across any number of cloud services too. Bringing this all together is more complicated than many businesses anticipate.

Data is Complex

The challenge here is that data is complex, and data delivery is complicated. While app design might look great, if users are having trouble with the performance of data searches, business transactions or general engagement, then the data integration strategy isn't right. With users' performance expectations growing and patience shrinking; revenue growth, customer engagement, and business operations are negatively affected.

The 5 V's of data complexity

1. Volume – the sheer scale of data we can access or need to access. Distributed across many services and systems.
2. Velocity – the speed at which data is being generated, the blurred lines between consumer and producer, and the need to move data around in realtime.
3. Variety – the ever growing differences between data resources (structured & unstructured), data models and data location.
4. Veracity – the unknowns, inconsistencies and fragmentation of data, that are making it increasingly difficult to harness.
5. Value – the inherent business opportunity that exists within and between data sets, and being able to unlock this value at the right time.

None of that complexity accounts for the biggest challenge – how to distribute important, relevant or realtime data back and forth over the Internet. And the Internet is a problem.

None of that complexity accounts for the biggest challenge – how to distribute important, relevant or realtime data back and forth over the Internet. And the Internet is a problem.

A short history of the Internet

The origins of the Internet date back to the 1960s, and the World Wide Web we know today began life in 1989. Over the last quarter-century using the Web, we've seen unprecedented growth, mass adoption and enormous amounts of data being transmitted over the Internet.

- Back in 2000, the total number of users on the Internet was 400 million. Today the Internet has almost 3.2 billion users, and China alone has 640 million Internet users.
- Annual global IP traffic will pass the Zettabyte (1000 Exabytes) threshold by the end of 2016, and will reach 2 Zettabytes per year by 2019.
- Globally, mobile data traffic will increase 10-fold by 2019 to 24.3 Exabytes per month, up from 2.5 Exabytes per month in 2014.

Customers, partners, employees – everyone – have come to expect more and more from the Web and from the Internet. This has prompted incredible innovation and investment from our network providers. One current project connecting London and Tokyo – 15,600KM of the latest high-speed fiber infrastructure – goes into service in 2016. The result of this $1 billion investment? Data delivery time between the two capitals is reduced by just 20ms – down to 168ms.

Put another way: an electronic trading platform can lose up to $4 million per millisecond if they are just 5ms slower than the competition.

Speed matters. While the Internet is improving, applications must consider how they maintain performance and offer interactive realtime experiences when it's struggling.

The Web was designed to move documents – not a continuous stream of data apps require. Data streaming has challenges – unpredictable and variable bandwidth, protocol limitations, frequent disconnections and reconnections, multiple device types, operating systems and versions, massive numbers of simultaneous users and unpredictable impact on backend business systems. All this means developers have a IoT to think about when offering fast, reactive and profitable apps.

What's needed is a way to help the Internet deliver data in support of interactive business apps. Apps need to collect information from all the relevant, disparate resources, cost effectively deliver that data over an inherently unstable Internet at scale and speed, and intelligently present it to end-users. And you have to do all of that without knowing:

- The speed of connection
- The connection's reliability
- The bandwidth availability
- Network restrictions imposed by the network operators
- The device, operating system, browser, and/or app version

Apps need to collect information from all the relevant, disparate resources, cost effectively deliver that data over an inherently unstable Internet at scale and speed, and intelligently present it to end-users.

That's just the broad-brush picture. Imagine the details and the underlying technology required to do this efficiently and effectively. Solving these problem apps require a data layer that can react and help overcome the Internet unknowns to deliver a realtime app experience that users will love. Because data is core to the app experience.

Solving Problem Apps with Data

Realtime data integration and delivery can set you on a path to the dream app you always wanted.

For problems outlined, realtime data integration and delivery can set you on a path to the dream app you always wanted. Push Technology has developed a unique approach to realtime data delivery we call Realtime Messaging. Born from the same requirements as Enterprise Messaging - integrating systems and apps that otherwise could not share data – Realtime Messaging was designed with today's cloud and mobile requirements in mind. With all integration use cases now relying on the Internet, the Push Technology difference ensures your data distribution platform reacts and delivers even when the network doesn't, and provides a reliable customer experience for your mobile apps.

Move away from traditional request/response data integration, and leverage realtime data integration.

Realtime data delivery for mobile apps – App developers need to move away from traditional request/response data integration, and leverage realtime data integration. This delivers better application experiences that react to business events as they happen and provides a bi-directional communication channel for application data.

Ensure a realtime data delivery solution offers:

- Event-driven integration, allowing applications to subscribe to the updates they are interested in and react to events in realtime.
- The ability to adapt to changing network conditions ensuring your web and mobile application experiences work – even when the network doesn't.
- Streaming bi-directional data protocols that support realtime engagement with the user, and enable collaborative applications.

If apps can leverage a reactive integration and abstraction layer, that hides the complexity of your data model, and decouple applications from potential backend changes then app developers can spend more time building features, and less time fixing integration problems.

Decouple & abstract backend systems from frontend apps – If apps can leverage a reactive integration and abstraction layer to hide the complexity of your data model and decouple applications from potential backend changes, app developers can spend more time building features and less time fixing integration problems.

Ensure you can decouple systems by:

- Providing a developer interface for legacy systems that don't natively support Web-friendly integration.
- Abstracting backend data models into a single, reactive data layer that protects applications from changes yet enables access to new data.
- Delivering unified visibility and management across all data assets, with granular access controls for each application.

Data backplane for Microservices Architecture – Microservices have emerged as the preferred approach to build scalable applications that can adapt as requirements change over time. In contrast to monolithic applications which typically have a single relational database, a microservices architecture requires a mechanism to manage data communication between a potentially large number of services that are often running different technology stacks and located across multiple cloud services. By using an event-driven data backplane that offers the SDKs your developers need, a given microservice can easily publish events that other services (existing or future) may subscribe to.

Ensure your data integration layer can:

- Have an event-driven integration model that promotes loose coupling between microservices, yet manages data consistency.
- Deliver a reactive data layer that adapts to data model changes, allowing topics to be dynamically created or destroyed without impacting each service.
- Serve your cloud-native architecture by optimizing data distribution between services connected over the Internet.

Conclusion

The Push Technology Realtime Messaging solution offers realtime data delivery, enables decoupling of systems, and supports the microservices architecture that tomorrow's digital products will require. By bringing all these features and best practices together, most problem apps can be overcome. Better yet, they can be turned into the dream app you always intended.

Our fundamentally different approach to streaming data – Realtime Messaging – is designed to overcome the challenges of the Internet. By providing simple unified SDKs for front and backend developers, Push Technology enables scalable realtime data distribution, while adapting to network conditions and dealing with disconnects. Available in the cloud or on-premise, our products Reappt and Diffusion will give your reactive applications a performance edge.

About Push Technology

We make the Internet work for our mobile-obsessed, everything-connected world. Leading brands like 888 Holdings, DAB Bank, IBM, and William Hill leverage our technology to power applications critical to revenue growth, customer engagement, and business operations. Learn how to deliver apps at scale and speed at **www.pushtechnology.com** or **www.reappt.io**.

We make the Internet **work** for the mobile-obsessed,
everything-connected world

LONDON
+44(0) 2035 880900

SAN JOSE
+1 408 780 0720

www.pushtechnology.com