Mobile Transformation
CTRIP’s Journey

Eric Ye @ ctrip.com

November, 2015
Ctrip - The Biggest OTA Platform of China

Online Hotel Booking Share
72%+

Countries and Regions Coverage
200+

Hotel Rooms Booking / Day
500K+

Airports Coverage
5000+

Peak Transport Tickets / Day
2 Million +

Peak Total Booking / Day
3 Million +
Problems:

- Offline: tech and operation
- Online: peers
- Mobile: infant
- Booking tools vs user experience
- Site scalability
Make Fast Changes:

- Open API
- Architecture 2.0
- Re-design Web UI 2.0
- Drive Traffic (SEO/SEM/…)
- Make product search fast
- Speed up Web Page
Mobilize Resources for App:

- Open API everywhere
- App catch up with web
- Create centralized mobile BU
- Make 1-stop travel App
Encountered Problems …
• Added 10+ new business lines
• App Lagged behind web functions
• Each BU fight for mobile resources
• App payment sucked
• Dev Infrastructure was at bottleneck

Strategy Shift:
➢ Decentralize mobile BU …
Multi-front Challenges solved:

- Re-organization
- PD must learn App fast
- Decouple everything (details later)
- Not to miss a single App release
From OTA to MTA
(Mobile Travel Agency)
Product  
Transform  
Tech
One App for Mainstream OS/Devices

Before

- iPhone
- iPad
- Android Phone
- Android Tablet
- Windows Phone

Now

Reduced
Everything Decoupled

- DB
- Servers
- App Release
- Infrastructure
- Business Unit
- Cost Unit
- Dev & Test
- Architecture
Problems seen in 2013:

- Totally coupled (monolithic)
- SPOF
- Lacked of LB routing
- No App monitoring
- Less secure
- ...

App Architecture: Monolithic
App Architecture: Microservices

Mobile Gateway

Hotel Service
Flight Service
Mobile APP Code: Modularized

- UI Components
- Business Modules
- Common Libs
- Mobile OS SDKs

Monolithic Project

- Data/URL Bus, Hybrid
- Foundation Framework
- Mobile OS SDKs
- Hotel
- Flight

Multi Projects

Vacation

Hybrid

Decoupled
Benefits of App Being Decoupled

01. Faster parallel development
02. Automate App integration testing
03. Faster App booting
04. Enable hot fixes
05. Increase function resiliency
App Networking Performs Fast and Reliable

- Use TCP and HTTP for App
- Fine-tune TCP for 2G/3G/4G/WIFI
- Tune retry logic at phase of connection/write/read

- Use IP list to avoid DNS failure and DNS hijacking
- Elect IP by its performance

- Reduce payload by protobuf & gzip
Mobile Networking Reliability

Achieved End-to-End Service Success Rate of 99.7%
Make App Perform Better

- **Faster**
  - App booting: load initial data & modules (only needed)
  - Data prefetching: initiate requests before page transition
  - Picture prefetching, compression and multi-level caching

- **Smaller**
  - Reduce App size with SVG
  - Prune ghost modules

- **Time to Market**
  - App hybrid design - Native & H5 coexist
Solutions to Mobile Maps Flaws

Flaws:

- Less accuracy on iOS Map
- Native Android LBS/Map is blocked
Enhance Mobile LBS/Maps Services

<table>
<thead>
<tr>
<th></th>
<th>Android</th>
<th>iOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>China</td>
<td>AMap Google Map/LBS</td>
<td>iOS Map AMap</td>
</tr>
<tr>
<td>Oversea</td>
<td>Google Map</td>
<td>iOS Map</td>
</tr>
</tbody>
</table>

Make App developers easy

- Unify LBS for iOS & Android
- Unify city locator
- Unify map services for Google/iOS/AMap
Mobile Monitoring at Runtime

End-to-End Monitoring:
- Service reliability
- Service time
- Networking metrics
- Crashing collection
- Consumption: battery, data
- Boot time
- Page time
More App Technologies ...

App Front
- Adaptive design
- Mobile payment
- 7z-compression

App Backend
- Voice search
- Push/Notification

- Watch
- TV
- ......
App for Apple Watch
High-level Transformation Best Practices

1. CTO to be best technological architect
2. Decouple App between domains as much as possible
3. Culture shift - Mobile first, everywhere and everybody
4. Let each PD team builds its winning App
5. App testing CI farm enables fast and quality App releases
6. Visualize and monitor App performance and quality
THANKS