~8 million tracks from Artist Labels High Level Architecture Diagram Production Environment – US (MediaNet) Amazon Web Services -MediaNet APIs pulls Stores Customer data, etc. audio tracks and metadata Amazon'Web Services Amazon Web Services **APIS** Yonder Prod Servers Yonder Production Portal –Web **AWS** Server In CO (for internal users) In CO **Production Production Backend Services Backend Services** Web-based Admin console which manages devices, user's data, 3rd party data (Charts) Android devices iOS devices Google Play - Yonder.apk App Store – Yonder Music

MediaNet Server – hosts audio files of

Labels – Universal, Warner, pushes content directly to High Level Architecture Diagram **Yonder Servers** Production Environment – US (Content from Labels) Amazon Web Services -Stores Customer data, etc. Content Ingestion - serves content locally from Labels Amazon\Web Services Amazon Web Services **APIs APIS** Yonder Prod Servers **AWS** Yonder Production Portal – Web In CO Server In CO (for internal users) Production **Production Backend Services Backend Services** Web-based Admin console which manages devices, user's data, 3rd party data (Charts)

iOS devices

App Store – Yonder Music

Android devices

Google Play - Yonder.apk

High Level Architecture Diagram Production Environment Celecom Malaysia (WIP)

Yonder Servers – Handle Scalability Celecom Servers -Celecom Servers – **Authorization** CMT, APIs **Production Backend Services Production Backend Services** (Json APIs), Authorization (Json APIs), Authorization **CELECOM APIS** Android devices – Yonder iOS devices enabled; Google Play -App Store – Yonder Music

Yonder.apk

MediaNet Server – hosts audio files of ~8 million tracks from Artist Labels MediaNet APIs pulls audio tracks and metadata Yonder Production Portal –Web Server In CO (for internal users) Web-based Admin console which manages devices, user's data, 3rd party data (Charts) Remove aws

High Level Architecture Diagram Production Environment – US

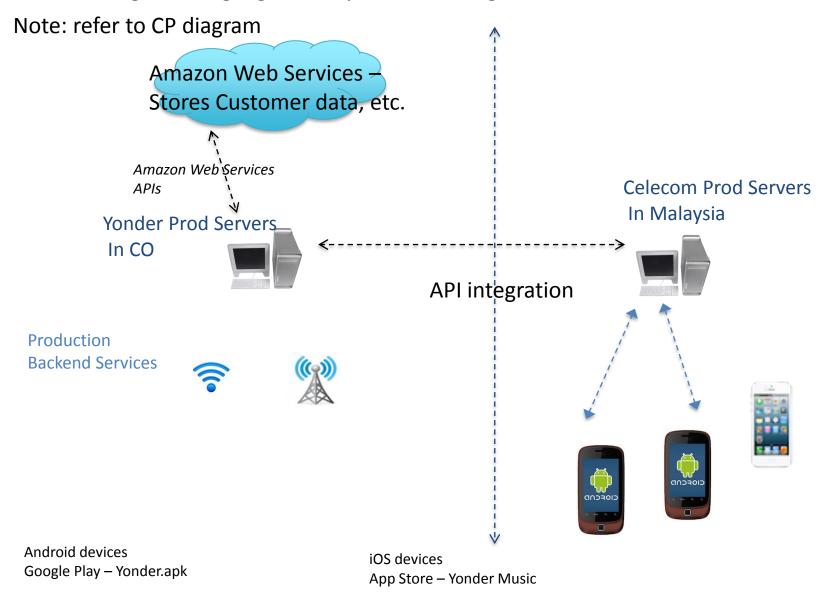
Google Play - Yonder.apk

Amazon Web Services -Stores Customer data, etc. Amazon'Web Services Amazon Web Services **APIS** Yonder Prod Servers **AWS** In CO **Production Production Backend Services Backend Services** Android devices

iOS devices

App Store – Yonder Music

Add protocols http 22 Wah carvicas Network Diagram – highlight APIs, protocols, integration level



Integration with Celecom – APIs Provides: (WIP)

- ➤ CMT Ringtones
- > Top Off (managing payment plans)
- ➤ Reporting features?
- ➤ Integrate on iOS and Android

Testing Goals: (WIP)

Launch Celecom-Yonder mobile app to customer base

Deliver a great user experience of app

Execute testing process to reduce defects, risk into mobile app for large customer base (1+ million users)

Testing Challenges:

Testing Scope:

Roles:

Responsibilities:

Testing Scope:

Yonder:

- Testing iOS and Android builds, reporting bugs
- Verifying existing and integration of Celecom features
- Working with CP in releasing builds in Staging, Production

Cardinal Peak:

- Performance Testing scaling customer base up 1K to 100K
- Low-Bandwidth using perfc tool
- API testing?, unit testing
- Testing Reporting Features from production portal?
- Releasing builds in Staging, Production

iOS Team:

- Provide iOS releases with new functionality
- Release builds in Staging, Production

Celecom:

- Celecom Portal Manage customer data, reporting?
- User Acceptance Test handoff to verify business functionality
 - Celecom integrated features
 - New/Existing users on Android, iOS
 - Low-Bandwidth
 - Online/Offline Mode
 - UI/UX; Notifications
 - Staging Environment

Testing Strategy (WIP)

How to show that we met our testing goals?:

- 1. Weekly Test Summary Reports show open issues
- 2. Traceability Matrix from requirements to test cases to show test coverage
- 3. Feature freeze, Code freeze -> Integration testing
- 4. Understand dependencies all layers
- 5. Release Notes for End user
- 6. Release Notes internally from Alpha Channel builds

Quality

Instilling Quality on these levels:

Product:

-Executing Smoke Tests, Regression tests

Process:

SDLC – managing defects, reduce risk

- -Requirements review for testability
- -Create JIRA dashboards for transparency into issues
- -Daily Summary reports to Product Owner, developers
- -Need Release notes for builds received

BEST Practices: User Stories Review, Root cause analysis

Test Estimation Effort for User Features:

Precondition: Staging envs are set up

Functional Testing (Android only):

Existing Features – 3 days each for 2 resources Security testing – 1 day each for 2 resources

Performance Testing:

(Low bandwidth) testing – 2 days for 1 resource Transition bet. WIFI, cellular, offline – 1 day for 2 resources

Integration Testing (via device):

- -API Testing
- -Ringtones: 5 days for testing, bug fixes
- -Top Off: 3 days for testing, bug fixes
- -Reports: 2 days each for testing, bug fixes for 2 resources

System Testing:

End to End – 2 days each for 2 resour

Exploratory Testing:

1 day each for 2 resources

Beta Testing in KL:

4 days for 5 resources (business user

User Acceptance Test in KL:

.5 day with Celecom business users

TOTAL: 20 days each for 2 resources; 20x8x2 = 320 man hours

Integration Tests

- Testing Celecom APIs
 - Ringtones
 - Top Off
 - Managing user plans (changing, renewing)
- Other APIS
- Reporting
- Testing APIs, via scripts?

System Level Testing

- End to End testing
 - From mobile devices across all systems
 - Admin console for Reporting (Celecom, Yonder portals)
 - Customer Support part
- Types of Testing
 - Functional Testing
 - Exploratory Testing
 - Non-functional Requirements testing

Beta Testing

- Beta testing
 - Test users using app as intended in local environment with business scenarios (day to day use)
- UAT testing
 - Handoff with Celecom business users
 - Traceability with requirements to UAT scenarios
 - Log all issues found

Launch Readiness Checklist:

- 1. Set up test accounts, test data in Staging Environment
- 2. Logging for Production events
- 3. Celecom feature integration
- 4. Full Regression test of new and existing functionality
 - a) Security testing
 - i. New Users/Existing users in Celecom
 - b) Performance testing
 - i. Low-Bandwidth issues
 - c) Localization App is in English
 - d) Functional Testing
 - e) Usability
 - iOS and Android consistent look and feel
- 5. Reporting Functionality
 - a) Reporting Portal Yonder
 - b) Reporting Portal Celecom
- 6. User Acceptance Test Signoff from Celecom
 - a) Report any issues found in UAT
- 7. Beta Testing completed

Deployment Strategy (WIP)

- 1. External Release Notes
- 2. Customer Support
- 3. Releasing Production Builds
- 4. Celecom Needs