

Video streaming & OTT testing Client Case Studies



Media Performance analysis, Test Automation and DevOps

- **Media performance Analytics**
 - **Client and Backend Test Automation**
 - **Application DevOps on AWS**
-
- Analyze and optimize media session's startup performance
 - Continuous test assurance with E2E Web(Platform), API and Streaming use case validation
 - Automated Testing of OTT Client Reference Applications
 - Continuous Integration, Provisioning, Deployment and Monitoring on AWS Cloud



Client:

Mediakind

Industry:

Digital Media Streaming

Technology Service:

OTT Client SDK

Scope:

Analytics to measure, analyze and optimize media session's startup performance

Challenge:

1. Capture media session's startup time with breakout for major events
2. Monitor and analyze performance data and identify areas for optimization
3. Generate performance report from analytics data as well as from device logs.

Solution:

1. Integrated trace module to measure time taken by identified major events and pushed the data to backend for each successful playback session
2. Automated scripts to run the performance tests and generate reports using device logs
3. Analytics dashboards to monitor startup performance

Outcome:

A tool to monitor media session's startup performance and to identify areas for optimization



Client:

Wurl

Industry:

Technology

Technology Service:

Content Distribution and
Advertising Services

Scope:

Continuous test assurance with E2E Web(Platform), API and Streaming use case validation

Challenge:

1. Requirement of WebUI(Platform, Admin etc.), API(Rest and GraphQL) and Streaming(HLS and OSMF) existing and new feature testing.
2. Seamless integration requirement with the CI/CD to automate the entire test suite on all platform.
3. Platform Web Video playback automation.
4. Streaming CLI automation.

Solution:

1. WebUI automation achieved using MABL tool.
2. Utilizing the OCR support to achieve the Video playback validation.
3. Postman and Mabl tool helps to test Rest and GraphQL API.
4. Robot Framework to cover Streaming related CLI test cases.

Outcome:

1. Using of MABL helped to execute test cases on demand on Cloud on different OS and Browsers. Even its ease of use helped to onboard new QA quickly into the automation efforts.
2. Web and API Automation helped to get feedback on the regression. Which reduced 2 weeks of Manual QA effort on each releases and focus on new features testing.
3. Covering streaming CLI test cases using Robot Framework helped in regression and focus on new features. Keyword based approach helped to onboard new QA quickly into the automation efforts.



Client:
Mediakind

Industry:
Digital Media Streaming

Technology Service:
OTT Client SDK

Scope:

Automated Testing of OTT Client Reference Applications

Challenge:

1. Automation of over 1500 manual OTT test cases for 4 different reference applications
2. Support platforms – Android, Android STB, iOS, tvOS, Edge, Chrome, Firefox and Safari
3. Playback quality analysis using playback statistics
4. Failure classification
5. CI-CD integration

Solution:

1. Adapted Rebeca's tried and tested 5G testing framework ABot for OTT use case
2. Adapt codeless automation and feature file reusability a core feature of ABot framework
3. Add support for various platforms using Appium and Selenium drivers
4. Capture playback analytics and validate playback quality
5. Rule based post processing of the results / logs for issue classification
6. Deployed in ADO for automated test execution

Outcome:

Highly adaptable ABot framework for OTT client player application with over 1500 reusable test cases



Client:

Wurl

Industry:

Technology

Technology Service:

Content Distribution and
Advertising Services

Scope:

Provision and Configure Infrastructure on AWS Cloud, Continues Integration and Deployment new Features, Extensive Monitoring Infra and Software and Alert on Failure

Challenge:

1. Create and configure infrastructure on AWS cloud.
2. Create/Update/Delete streaming Live and VOD channels along with EPG (Electronic Program Guide). Create/Maintain cron jobs for creating and delivering streaming video contents and EPG.
3. Create CI/CD pipeline to support quick testing and delivering software into production.
4. Monitor infrastructure and software so that failures can be captured before client experience that.
5. Alerting on infrastructure and software failure.

Solution:

1. Used Terraform and CloudFormation Stack for provisioning the infra on Cloud. Used Ansible for configure the infra.
2. Used Jenkins for CI/CD pipeline integrated with Ansible. The Jenkins pipeline had used to provision/update/delete streaming channels and EPG along with creating and delivering video contents and EPG.
3. Used Datadog Monitoring tool integrated with PagerDuty for alerting.

Outcome:

1. Terraform/CloudFormation Stack, Ansible, Jenkins Groovy files are checked in GIT so that engineer can version it and can be used by across the team.
2. Used CloudFront for content delivery which gives faster delivery and low latency so customer can experience faster seamless streaming.
3. Used Grafana to visualize all real time channel information from DB.
4. Using Datadog Monitoring tool give you setting up monitoring with Graphical view of infra. It easily integrates with alerting tool PagerDuty which sends alerts at failure.