Enhancements in RBS Tech

Project report submitted in fulfilment of the requirement for the degree of Bachelor of Technology

In

Computer Science and Engineering

By

Prakhar (131333)

Under Supervision of

Shubham Agrawal as **Manager,** &

Lalit Pardhi as **Mentor**

To

Department of Computer Science & Engineering

Jaypee University of Information Technology, Waknaghat, Solan-173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report regarding various Enhancements done in RBS Tech, in fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from February 2017 to August 2016 under the supervision of **Shubham Agrawal (Reporting Manager).**

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Prakhar (131333)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Shubham Agrawal

Software Development Manager III (L-6),

Amazon Development Centre,

Bengaluru

Dated: 11 Sept, 2017

Acknowledgement

It is my privilege to express my sincerest regards to my manager, Shubham Agrawal, my team mates, Lalit and Murali for their valuable inputs, able guidance, encouragement, whole-hearted cooperation and direction throughout the duration of our project.

I also thank faculty members and my department's head, **Prof. Dr. Satya Prakash Ghrera**, for guiding me to put forward each step and being able to drive and implement corporate projects with great efficiency without losing my calm under critical situations.

I deeply express my sincere thanks to my Team RBS Tech for having faith on me and providing me challenging works, I have accomplished.

At the end I would like to express my sincere thanks to all my friends and others who helped me directly or indirectly during my internship at Amazon Development Centre.

Date: 11-09-2017 Prakhar (131333)

Table of Contents

Sr. No.	Topics	Page no.
1	Introduction	1-2
2	Project 1	3 - 7
2.1	Introduction	19-22
2.2	Problem Statement	23-24
2.3	Overview	25-31
2.4	System Design • High Level Design • Low Level Design	32-33
3	Project 2 : EU solution for AVS	8-21
3.1	Problem Statement	8
3.2	Introduction	8
3.3	Survey	8
3.4	Model Design • Organizational Model	8
3.5	Beta implementation for AVS organizational model	10
3.6	Benefits of AVS	14
3.7	Appendix • Appendix I • Appendix II • Appendix III • Appendix IV	17
4	Project 3	
4.1	Introduction	40
4.2	System Components	40
4.3	PCS Platform	41-46
4.4	PCS Platform - Support for Debugging • LLD	
4.5	Admin Update Page • Problem Statement • Solution Approach • DESIGN HLD LLD	33-35

5	Project 4	37-48
5.1	Problem Statement	37
5.2	Requirements & Objectives	38
5.3	Architecture • High Level Architecture	39
5.4	Components & Flow of Request	43
5.5	Steps to develop a new automation	48

Introduction

RBS (Retail Business Services) was started in 2008 to support the Retail teams worldwide. This team is primarily responsible for 4 core tasks:

- New Item Setup (NIS),
- Andon Cord resolution,
- ISS ticket resolution, &
- Problem Receive issue resolution.

New Item Setup: In this process, RBS receives product details from vendors, mainly through two channels: Vender Central and GIM. RBS "scrubs" the provided details, completes the missing information and converts this data into Amazon standard formats (as per the standard guidelines in different categories/product lines). After this, RBS sets up these items in Amazon systems by creating new ASINs along with creating appropriate relations with new/existing ASINs.

Andon Cord: Andon Cords occur as a result of customer contacts that have indicated that there could be a serious defect with the product. As a result, Customer Service (CS) will halt sales on an item by pulling the Amazon Buy box down from the site. This deters additional poor customer experiences and allows RBS time to investigate the potential customer-facing issue. RBS works to resolve these issues with the assistance of the vendors and retail business team.

When a CS representative notices a potential serious defect in the product being shipped out, she/he initiates an Andon Cord. At this time, the Buy box is suppressed and the ASIN is now not buyable. The goal is to prevent future customers from buying an ASIN that has a CS-identified defect trend. In addition, RBS will immediately provide all available information to the appropriate retail team so the root cause can be identified as quickly as possible.

ISS - Inbound Support Service: ISS refers to a collection of trouble tickets that are brought to RBS's attention by FC / non-FC associates. ISS is a process to solve the receive problems that are caused due to several reasons; (vendor initiated, incorrect catalog quality, etc...). ISS is handled through a common interface, Trouble Ticketing, where the FC associates and the retail teams communicate. Each issue will be addressed by the FC associates / Non-FC who files a Ticket through a common interface Remedy Ticket. When there is an exception/ discrepancy between what was ordered by the buyer and what was received into the FC, an Inbound Support Services (ISS) ticket is created by the FC.

Prob Rec - Problem Receive: Problem Receive refers to inventory received into the FCs that have immediate issues or problems. These problems can range from extra units being shipped to incorrect product being sent to an Amazon FC. This process entails the FC communicating such problems to RBS, RBS identifying the issue and resolving for FC receipt, vendor return or liquidation. Each Receive Problem Item (RPI) can exhibit one or more Receive Problem Types.

Active types are:

- NOT ON PO - Items are sent to Amazon by a vendor to Amazon and are marked by a vendor as being on a PO. These are Prob Rec because the items are not on the PO that Amazon sent to the vendor.
- UNCONFIRMED (introduced Oct 9, 2007) - Items are sent to Amazon by a vendor that were ordered on a PO. These items were not confirmed prior to shipment to Amazon.
- CANCELLED - A vendor ships items that were cancelled on a PO.
- OVERAGE - The number of units of an ASIN received exceeds number of units confirmed on the PO. Hence, an appropriate PO is chosen to accommodate the excess units of this ASIN and then it is received by the FC.
- WRONG FC - The vendor ships items to the wrong FC. These units are received under problem receive and then transshipped to the correct FC as required.
- VENDOR DAMAGED - Items shipped by the vendor are damaged. Hence, these items are received as unsellable inventory and returned back to the vendor.

RBS Tech focuses on turning advanced human workflows into sophisticated automated systems. The systems developed by RBS Tech have a direct impact on customer buying decisions and online user experience. Team owns and builds solutions for New Item Setup and Defect Reduction for Amazon Retail. Vision of RBS Tech is to build solutions to automate systems and processes to eliminate human touch points from New Item Setup, ISS Defect Resolution, Andon Defect Resolution and Flex task owned by RBS.

During my internship at RBS Tech, I have worked successfully on following areas:

- Design and Implementation of EU solution for AVS
- Development of peer recognition tool, Cheer-A-Peer, &
- PCS Solution and other RBS Enhancements

Project 1: Cheer-A-Peer Tool

Introduction

This end to end project was developed for peer recognition in the team. Cheer-a-Peer Tool provides a platform where peers can appreciate each-other for exceptional performance on Amazon leadership principles.

Cheer-A-Peer tool was originally built for RBS Tech, CMT Tech and DE Team to recognize peers.

Problem Statement

Every individual wishes to be recognized whenever he has performed beyond his "Call of Duty". Appreciations must not always happen from top to bottom in org hierarchy, which means the appreciations should not be always coming from the manager. To motivate the thought for peer recognition, a tool with simple User Experience was required, which was not already present in Amazon System.

Overview

Cheer-A-Peer is a Web application which provides a simple interface to send Recognition/Cheer (certificates as e-mail) to the desired employee and copy the whole team in the mail.

Apart from sending e-mail notifications, it has its own audit trail mechanism which stores the history of all appreciations to be viewed at Manager level.

Cheer-A-Peer application has two main components -

- Send Cheer to peer.
- Reports at Manager Level.

Cheer-A-Peer web application consists of four pages -

- **Home Page:**
  The home page is visible to all the employees and it contains link to rest of the pages -
  a. Cheer Page
  b. Report Page
  c. FAQ Page

- **Cheer Page:**
  The cheer page is used to send the e-mails to the peers. The cheer page contains input field for employee Id, recognition and some message to be sent in e-mail (as certificate) to the peer. This is a restricted page and only users from on-boarded LDAP groups will be able to access the page and send cheers.

- Following actions are performed after opening the page:
  - Employee Login Id is extracted using the Sentry Service.
  - User is validated by checking the current login Id in login Ids present in the on-boarded LDAP groups for different teams.
  - If user is valid then user is allowed to access the page else, redirected to Request Access page.

- Functionality provided:
  - Accept awardee Id, recognitions and message to be sent in certificate.
  - Validate input provided by the user and shows corresponding errors message.
  - Preview the certificate that will be sent as the cheer.
  - Send the cheer as e-mail to the awardee and sender's team as well.

- **Report Page:**
  The report page is used to get the cheers at the manager level. It accepts the managerId and displays the information of all the cheers received by the employees directly under the given manager. This is a restricted page and only valid users are allowed to access the page. At first it just displays the count of the cheers received. But, user get detailed information by selecting particular employeeId.

- **FAQ Page:**
  The FAQ page is a public page and contains answers to basic questions related to the Cheer-A-Peer Application.

This tool is designed to on-board new clients very easily. The tool authenticates users that belong to the on-boarded LDAP groups. The LDAP groups with access are stored in the configuration file as shown below.

```
#####   on-boarded ldap groups   ######
*.*.ldapGroup = ("rbs-dev-core","rbs-cheer-a-peer-integ","cmt-tech");

###### cc e-mail for particular ldap group #####
*.*.ldapToCCEmailMap = {
    "rbs-dev-core" = "cheer-a-peer@amazon.com";
    "cmt-tech" = "cheer-a-peer@amazon.com";
};
```

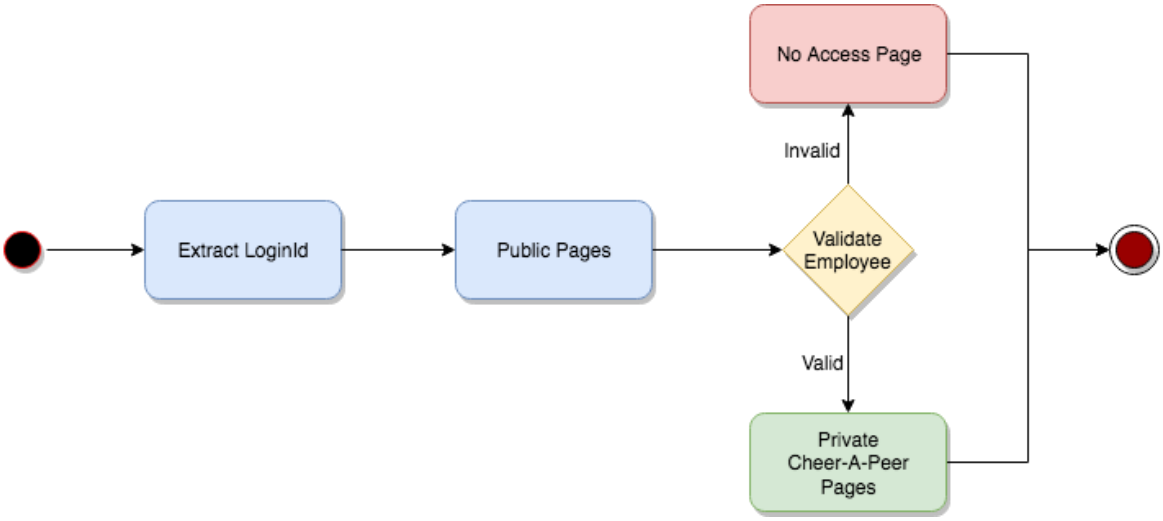
If user is valid, the LDAP group is stored as a session variable to be used for further use. This is because recognition emails need to be sent to team members of awardee only and not to other teams. Hence, while sending notification e-mails, cc e-mail alias will be chosen based on the LDAP group of the user.

Onboarding other teams on to the tool:

To on-board new team, we just need to add the LDAP group of the new team into the configuration file and along with the cc e-mail alias corresponding to the new LDAP Group. That's it.

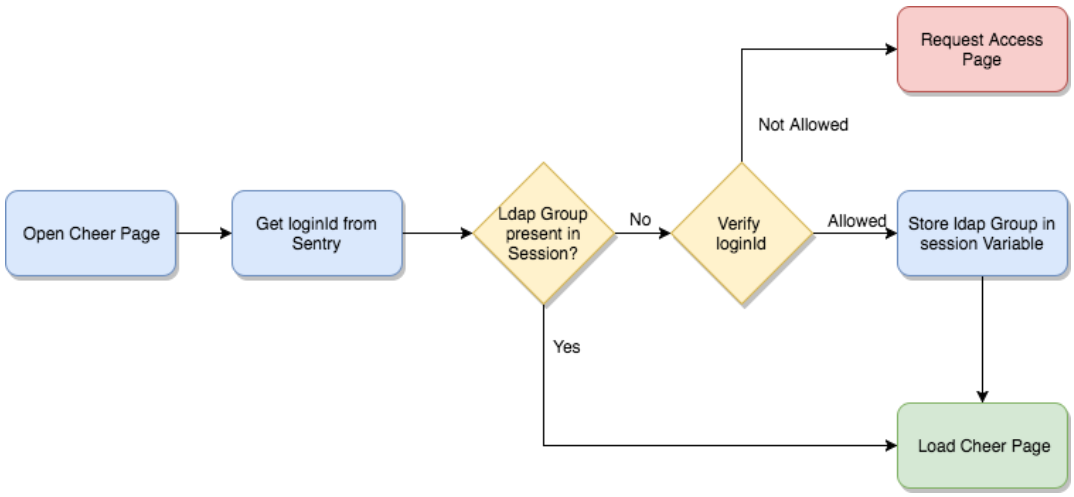
High Level Design

- Web Application Flow:

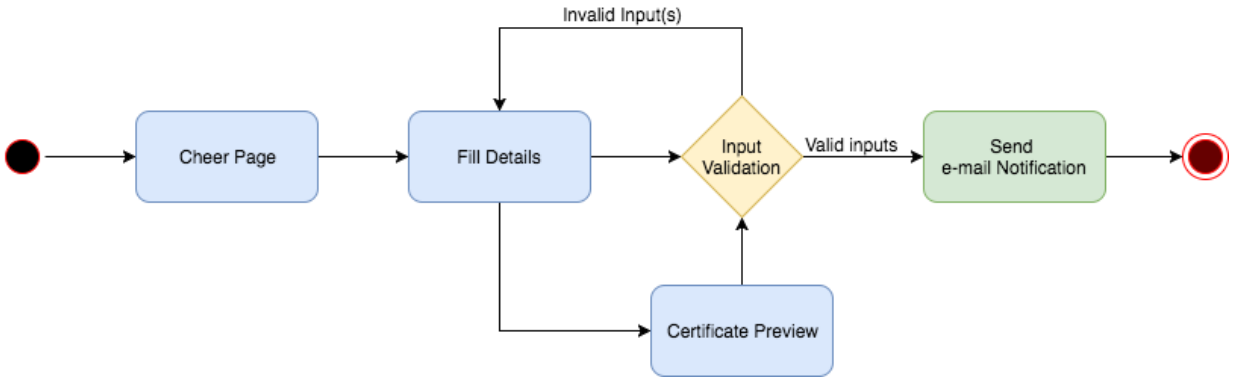


We validate users before redirecting them to private pages. After sentry authentication, the user Id is fetched from HTTP request header. This User Id is passed to the validator, where it is checked whether the logged in user is the member of allowed LDAP Groups or not. Whether should the Ldap group be given access to the private pages, depends on whether the team is our client or not.

- Cheer Page Flow:



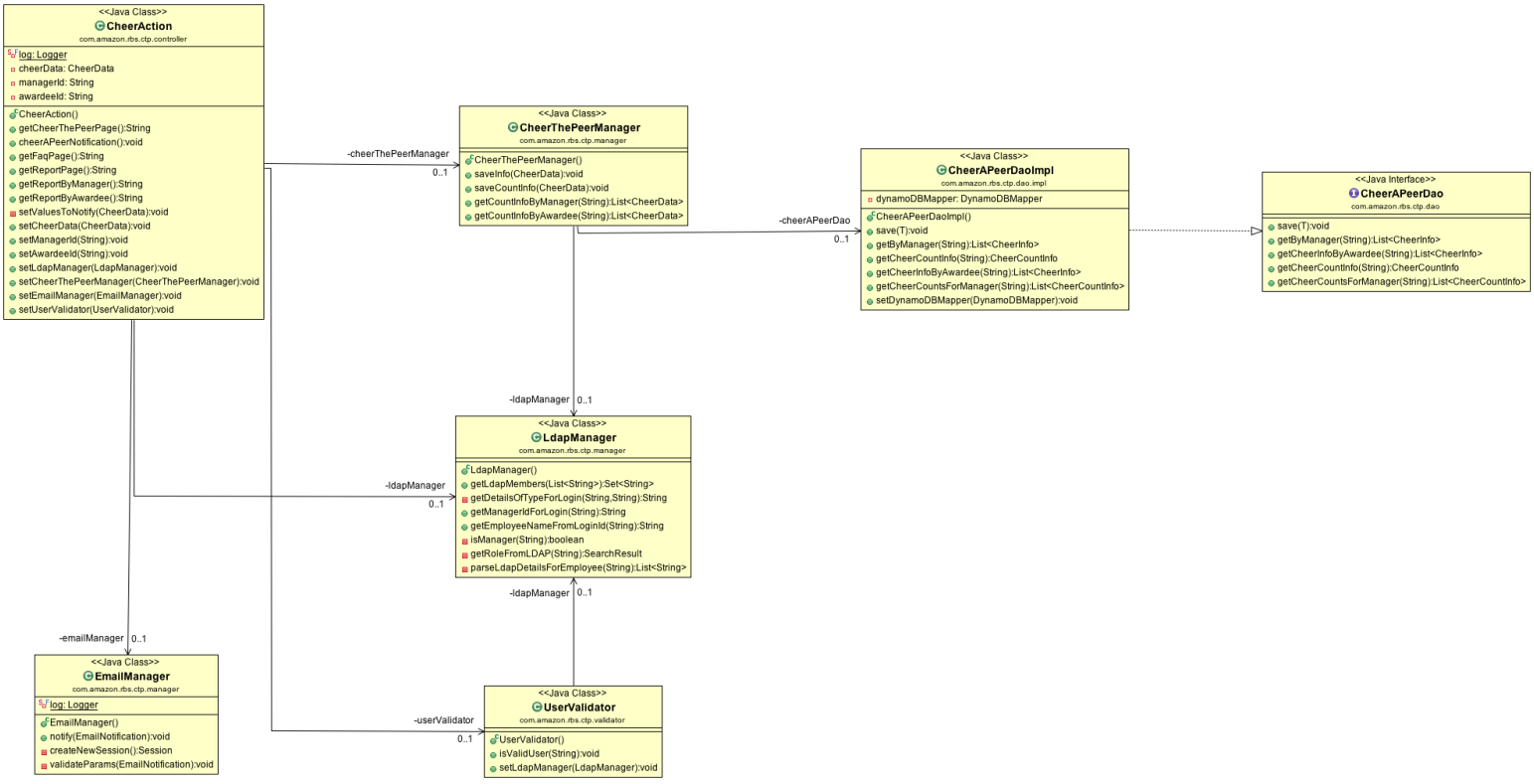
- Cheer Flow:



Once validated, the user has access to the cheer page, where the input fields are to filled for the peer recognition. The user can preview the certificate on filling input fields. On submitting the form, validations using JS take place on the front end. If the input fields are valid, request goes to the controller, where again the basic request validation takes place. Now the sender details are set and an email notification is sent, followed by saving cheer info to the Dynamo Db table.

Low Level Design

I have followed Struts2 MVC framework on Spring platform, where I have structured the package in a way to have View developed by JSP pages,



Our Clients

This employee recognition tool was brought up for promoting peer recognition in RBS Tech and CMT Tech teams. But, the flexible design permits on-boarding many more clients in future. This application is designed to have separation for different clients, ie. an employee recognition for one client team will not send notification to the email-alias of other.

The developer, in order to add a client, needs to just add LDAP Group and corresponding email-alias to the config.

To onboard to Cheer-A-Peer, one needs to reach out to us at cheer-a-peer-admin@amazon.com.

Our clients today are:

- RBS Tech Team
- CMT Tech Team
- RBS Program Team

Project 2: EU solution for AVS

Problem Statement

Currently SVS Brand Specialists do not log or update any of the activities conducted on behalf of their vendors. This information gets lost in multiple, scattered Excel files, SharePoint or paper sheets. At the moment there is no defined template or Data Model defining how this data is to be stored.

Introduction

In November 2016, Amazon Vendor Services (AVS) was launched to enhance current Strategic Vendor Services (SVS) program with the main goals of

- increasing vendor satisfaction, &
- making the program scalable.

Survey Analysis

According to Amazon's most recent time tracking survey, Brand Specialists spend 33% of their time on “non-SVS” work (details by Division and local in **Appendix IV**). 21% of Brand Specialists, typically the most senior ones, spend over 50% of their time on “non-SVS”. This work covers a broad scope:

- 29% is VM-type work,
- 19% is project work,
- 19% is Site Merch work,
- 14% is SME/POC work, and
- 9% is In-Stock work.

Finally, 44% of the time spent on non-SVS work can be allocated to activities in the AVS service menu (but for non-SVS vendors) and will benefit from solutions being developed by AVS, while the remaining 56% will need to be addressed separately.

Task level details available in **Appendix IV**. In order to have AVS Teams focus 100% on AVS vendors (see FAQ 4), we will separate GL-related tasks performed by Brand Specialists from SVS-related tasks as we move forward to the new AVS organization, by leaving temporarily 20% of the SVS HC within the GLs to take care of these activities.

The approach of separating GL related tasks from AVS tasks presents several advantages:

- it is a clean organizational cut with clear accountability that will enable high visibility on what people are doing and will prevent GLs from using Brand Specialists as a pool of free resources;
- it will allow Brand Specialists to have clear goals and priority set by their managers rather than being split between AVS Team and GL work;
- positions left in GLs can be leveraged as mitigation for VMs losing directs;
- Category Leaders and VMs will be incentivized to optimize current activities.

Organization model

In order to mirror the retail organization, teams of Brand Specialists, organized by GL when possible and led by L5 Team Leads, will report into L6 AVS Managers operating within the boundaries of retail Directors and reporting into these Directors or one of their designated Category Leaders. Brand Specialists will continue to be POCs to their vendors, and therefore also to the VMs managing these vendors. Moreover, Brand Specialists will sit next to the VMs as they do today. The Team Leaders, typically more senior Brand Specialists (L5), will be spending part of their time managing AVS vendors and the other part managing their team and being the POC to their Category Leader(s). These teams may also have experts working across vendors/categories, typically on advanced operational efficiency and traffic activities, and either reporting into Team Leaders or AVS manager, depending on where the vendor demand is (in one GL or across GLs). This model will preserve i) a seamless communication between VMs and Brand Specialists, ii) the ability to align on priorities and resolve conflicts and iii) access to the category information required by Brand Specialists to do their job. At the same time, it improves our ability to i) effectively drive the standardization required to reach our service quality and productivity goals, ii) develop expert profiles as required by vendors and allocate them to the right vendors, and iii) manage business variability without any service disruption.

While we will seek to roll out this "mirror" model to the extent possible, certain considerations need to be taken into account. AVS Manager organizations will need to be large enough to cover at least 40 vendors which is the critical mass required to smooth business variability and ensure that experts are always utilized. Also, they should not exceed 30 HC for internal manageability purposes (AVS Managers should have no more than 5 Team Leads who in turn should have no more than 5 directs – preferably 4 since they are also managing their own vendors). Finally, divisions with multiple AVS organizations will need to merge them when they jointly have less than 25 HC, in order to minimize overhead (no merging of organizations across divisions). Details around this framework can be found in **AppendixVII**. While IT & ES will have joint AVS organizations, we need to avoid having L4s reporting into remote managers; this means that we will need local Team Leads working across GLs in both countries (rather than working for GLs across countries). We still need to define how the very small divisions (Fresh, Digital, Auto/BISS) will function, and to deep dive on Media/ Digital, which is offering different services to SVS vendors and in UK has already moved to a pooled organization (ETA for proposal end of mid-May).

In order to simplify the transition, we have made an initial organizational proposal by marketplace, with more flexible rules (i.e. 30+ vendors per pool rather than 40+; and no hard requirements to merge pools which jointly have less than 25 HC to reduce overhead). See **appendixVIII** for proposed deployed organization for each marketplace.

Finally, marketplaces will need to designate one of the AVS managers as SME to the central AVS team, who on top of his responsibilities will be required to gather requirements from his local peers, contribute to the development of tailored solutions and guarantee full adoption of technology, HOTW and offshore processes. In order to ensure that all country and division voices are heard we propose the following SMEs:

1 from DE Other Hardlines, 1 from UK Consumables, 1 from FR Media, 1 from IT-ES Hardlines, and 1 from Softlines.

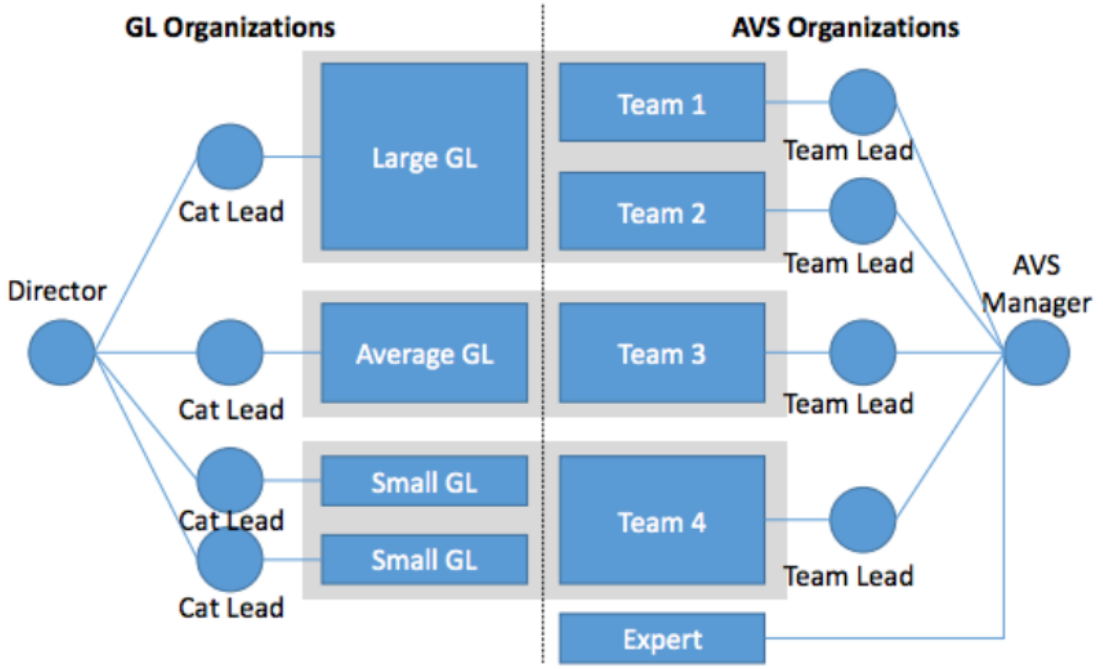


Fig. Illustration of the Mirror Organizational Model

Beta implementation for AVS organizational model

A group of GLs (DE PC, Home and Furniture; UK Beauty, CE, Wireless, Camera, Home Entertainment; FR PC, Office, Consumable, HL2; IT/ES TBD; Softlines Watches & Jewelry) will start transitioning to the new AVS organization model straight away, as soon as they hire their AVS Managers. For these GLs, we have developed recommendations on how AVS teams should work, defined roles and responsibilities, proposed interaction models between Category and AVS teams, and suggested goals for the AVS teams (we will provide this as a separate document). We will leverage these GLs as a privileged community to test automation and offshoring. Some of these organizations will move immediately to the final target set-up (see **Appendix III**) and will be leveraged as "Beta implementation".

The Large GL Average GL Small GL Small GL Team 1 Team 2 Team 3 Team 4 Expert Cat Lead Cat Lead Cat Lead Cat Lead Director Team Lead Team Lead Team Lead Team Lead Team Lead AVS Manager GL Organizations AVS Organizations Fig. 1 – illustration of the mirror organization model AMAZON CONFIDENTIAL 3 goals of the "Beta" will be:

- to fine tune these recommendations and learn how AVS Teams can best adopt new AVS tools and processes,
- test new tools and offshore services, &

- identify potential friction points with vendors, employees and the GLs so that we can react and provide recommendations to other GLs.

In order to achieve that we will put in place a set of mechanisms that will include:

- weekly interactions with the AVS central team to deploy standardized processes and tools,
- monthly monitoring of key metrics which would trigger alerts, &
- periodic interviews with employees involved.

Feedbacks are being gathered from Category Directors of Beta GLs in order to finalize mechanisms before Beta starts (ETA Mid-April). These mechanisms will remain focused on the impact of the organizational model, which is the object of the Beta. In parallel, we will continue to monitor overall vendor satisfaction through 2 surveys per year, and we will build dashboards to measure the impact of AVS on core vendor metrics (ETA end of Q3).

Communication plan for Beta Implementation

We will not proactively communicate AVS implementation to minimize churn on employees. However, we will provide M-teams with FAQs so that they are updated and have talking points on the "direction of flight".

AVS general directions will also be communicated in the call of Country Managers on Future of Category Leaders planned on March 31st. Specific sessions will be held with employees participating to Beta implementation, namely:

- Preparatory meetings with Category Directors and Category Leaders supported by AVS central team (additional FAQs on the mechanics of AVS Team work will be provided)
- All hands with Division Leads, Category Directors and all impacted employees supported by AVS central team and local HRBP
- In the week following the all hands, Category Directors will have 2:1 with VMs and Brand Specialists and their managers

AVS eligibility tenets

Tenets for AVS eligibility (unless you know better ones):

- We want AVS to remain accessible to the largest base of attractive vendors, attractiveness is a mix of positive contributions to CX and economic viability defined at GL level.
- We will keep AVS eligibility as an incentive for vendors to become more attractive to Amazon in Marketplaces and/or GLs where negotiation leverage is needed.
- AVS eligibility criteria must be easily understood by vendors and must be consistent for vendors that operate across multiple marketplaces or GLs.

Accordingly:

- Services that improve Selection & CX are available to all vendors

- Services that primarily accelerate vendor growth are available only to attractive vendors, or vendors that commit to a clear auditable plan to reach attractiveness within a short timeframe (or attractiveness above average in GLs where most vendors are structurally not attractive)
- The most valuable services that benefit the vendors are available only to vendors that are accretive to GL attractiveness, or vendors that commit to a clear auditable plan to become accretive within a short timeframe.

Next steps are to convert these tenets into specific eligibility criteria. This will allow us to estimate increased entitlement of the service (i.e. total number of vendors eligible under new criteria) as well as CCOGS at risk (i.e. current SVS/PCS vendors that will be rejected from AVS Level 3/2 under new criteria). We will discuss these criteria and resulting business impact in our meeting with Russ scheduled on May 30th.

Risk connected

AVS is designed to positively impact vendors' satisfaction and renewal rate, as it will allow to diagnose their needs, define relevant action plans, monitor business impact, improve the quality of the service delivered, and provide new highly value services.

However, we will monitor impact on vendors' satisfaction and renewal rates. We expect to detect any potential red flag related to AVS transformation by tracking renewal rate of agreements terminating in Q3 and Q4'17 and to be able to react on that before Q1'18. These agreements represent 18% of the total, corresponding to €8.7MM profit.

Alternative Approach Analysis

One of the most hotly debated topic is whether or not to leave the GL related work done by Brand Specialists within the GLs, or to give the flexibility to pool it within the AVS organization.

Pro's to pool GL related work within AVS organizations are

- to keep interesting projects / SME responsibilities within AVS and add attractiveness to the positions,
- accelerate GL tasks scaling as quarterly targets will be applied and we can start more quickly not to backfill some of current Brand Specialists, &
- to benefit from the AVS Managers' help to scale the GL-related tasks that are similar to AVS tasks (44% of GL related tasks are part of the AVS service menu according to time tracking survey – we are aware that there is a blurred line between some AVS tasks and GL tasks, which we will take the point to discuss with local teams to make sure AVS menu is comprehensive and without grey areas).

Cons are

- unclear accountability on GL activities (Brand Specialists will end up having two managers, one for AVS and one within the GL), and this will be augmented by grey areas between AVS and non-AVS tasks;

- missed opportunity to leverage positions left in GL as mitigation for VMs losing directs;
- lack of clear visibility on GL activities done by Brand Specialists, increasing the complexity of tracking their reduction and the risk of maintaining AVS as a fall-back bandwidth option for additional GL tasks; &
- AVS Teams will have to focus also on the 56% of GL activities which are not in the AVS service menu, for which they are not fully accountable, leaving category Leaders and VMs less incentivized to optimize these activities.

AVS US

The US vision for SVS is aligned with EU, however implementation roadmap is different.

The US is creating a central team of Vendor Specialists. These Vendor Specialists will focus on vendor account management, with work executed by centralized functional teams (RBS), similar to the EU and will report into Retail Systems organization.

New SVS vendors are supported by the central team. The new model is being piloted in 5 GLs during H1 2017 and then will be rolled out to the remaining GLs.

At this point, Brand Specialist positions within the GL will cease being backfilled and as existing Brand Specialists rotate, vendors will be transitioned to the support of the new central team of Vendor Specialists. At the same time, work previously executed by Brand Specialists will be transitioned to the centralized RBS team. This will naturally lead to a gradual elimination of vendor-funded resources dedicated to GL activities. A dedicated workstream led by a Category Leader is mapping these activities and assessing what the new category structure looks like in the light of the many automation initiatives.

AVS different from SVS

Strategic Vendor Services (SVS) was designed to help our best vendors grow on Amazon thanks to focused attention from Brand Specialists. Amazon Vendor Services (AVS) will enhance the current suite of products (SVS, PCS and PSS) and structure it around a menu of ~80 standard service modules (see **Appendix I**). We will launch a new platform called Amazon Vendor Services Workflow that will support VMs and Brand Specialists to diagnose vendor needs, select the right services, define and share with them an action plan, and transform it into an operational plan of tasks to be executed over the duration of the agreement. Each service module will have a playbook and SOPs to ensure quality and consistency of the service provided and will be mapped to output metrics, so that we will be able to track and report business impact to vendors. We will automate and offshore to RBS (Retail Business Services, Amazon internal department) basic and standardizable tasks, so that Brand Specialist will have more time to focus on vendor facing and high-value activities. This will allow each Brand Specialist to manage more vendors than today and to scale up SVS organization, which is not sustainable with the current set-up (more than 930 vendors and 600 Brand Specialists steadily growing).

The service modules are split along 3 levels of service according to their added value and complexity to deliver, with the highest level (Level 3) only accessible to our best vendors. “Level 3” services require an understanding of the vendor’s business, a direct relationship with the vendor and tailored planning to be effective and hence will be supported by Brand Specialists. “Level 2” services require light onshore support to gather vendor requirements and tailor the plan accordingly, but are mostly executed offshore and through automation. “Level 1” services are totally offshored or automated.

AVS Team organization

An AVS team is defined as a group of Brand Specialists reporting into an AVS Manager (typically L6) responsible for service quality, productivity and growth, and detached from day to day vendor activities. Brand Specialists continue to be POCs to their vendors and will be organized into sub-teams managed by more senior Brand Specialists (typically L5) spending part of their time managing AVS vendors and the other part managing their team and category stakeholders. This structure will mirror the retail organization i.e. Category Leaders will typically have an L5 team lead as POC while VMs will have designated Brand Specialists as POCs (see proposal in Appendix III). These teams may also have experts working across vendors/categories, typically on advanced operational efficiency and traffic activities, and either reporting into sub-teams or into the AVS manager, depending on where the vendor demand is. Finally, the AVS managers will report into the retail division - reporting lines to be defined by division heads.

This model will preserve i) a seamless communication between VMs and Brand Specialists, ii) the ability to align on priorities and resolve conflicts and iii) access to the category information required by Brand Specialists to do their job. At the same time, it improves our ability to i) effectively drive the standardization required to reach our service quality and productivity goals, ii) develop expert profiles as required by vendors and allocate them to the right vendors and iii) manage business variability without any service disruption.

Teams will need to be large enough to cover at least 40 vendors which is the level of aggregation required to smooth business variability and ensure that experts are always utilized; and should not exceed 30 HC for internal manageability purposes. Finally, divisions with multiple teams will need to designate one of the AVS managers as SME to the central AVS team, who on top of his responsibilities will be required to gather requirements from the entire division, contribute to the development of tailored solutions and guarantee full adoption of technology, HOTW and offshore processes within his division.

Benefits of AVS

What benefits does AVS bring to vendors?

- AVS proposition is based on a set of over 80 standardized activities (see **Appendix I**) from which the most relevant ones can be selected according to vendors’ needs. Brand Specialists will have access to training, playbooks and

SOPs for all AVS activities. As a result, we will be able to deliver a consistently higher quality of service.

- AVS Workflow technology will support VMs to analyze their vendors' core business metrics, define relevant activities, and propose a tailored plan to reduce/ close their performance gaps. Vendors will receive frequent progress updates on activities and their impact on business performance.
- Vendors will also have access to dedicated advanced services (e.g., VLT improvement). As Brand Specialists re-organize into centralized teams, some of them will have the opportunity to develop highly specialized skills and expertise and provide advanced services to all the AVS vendors supported by the team.

What benefits does AVS bring to Amazon?

- Vendor satisfaction and consistently higher quality of service.
- Scalability. The current set up of SVS is not scalable, as it requires HC to grow almost linearly with vendors. With already 550 HC this is a challenge in term of recruitment, people development, standardization of the service and makes it hard to ensure consistency and quality of the service. AVS will allow increasing Brand Specialists productivity (vendors served per HC) by offshoring and automating basic and standardizable tasks. Centralization of Brand Specialists will enable monetization of savings
- Flexibility. Centralization of Brand Specialists will enable higher flexibility in the support we provide to vendors. For example, we will be able to provide more support to a vendor during a given period if needed, and to start a new agreement immediately without waiting to hire a new dedicated Brand Specialist.

What benefits does AVS bring to employees?

<u>I'M A BRAND SPECIALIST:</u>

We expect the Brand Specialist job to become more attractive than today. You will spend less time on "back-office" activities, which will be offshored to RBS or automated, and will spend more time facing the vendors or performing high-value activities. While you will still be part of Retail organization and have opportunities to proceed your career as VM as you have today, you will also have new development opportunities. In fact, some of you will have the opportunity to grow and lead teams of Brand Specialists, and others will develop specific functional expertise and deliver advanced services to vendors.

<u>I'M a VENDOR MANAGER/ I MANAGE a BRAND SPECIALIST:</u>

AVS is an enabler to your role as a Vendor Manager, as you will get access to a more powerful system to help you deliver. AVS workflow will support you to identify potential vendors for AVS, assess their needs and improvement opportunities so that you can effectively pitch the service to them, select the right services and build and drive impactful action plans. AVS workflow will provide you with automated reports and standardized recommendations saving you time and improving the effectiveness

of your work. It will allow you to track progress on your action plans, adapt it to your vendors' evolving needs and monitor their performance in real time.

Centralization of Brand Specialists

Centralizing Brand Specialists will allow us to:

- Drive adoption of standardized, automated and offshored processes thanks to a small number of AVS managers highly accountable for this and closely supported by the central team.
- Provide the business stability required to forecast vendors and HC in advance, ensuring service continuity.
- Deliver advanced services to any vendor through specialization of specific profiles within the team.
- Turn time saving initiatives into productivity improvements fast (e.g., saving 5% time in a team of 20 through offshoring allows to save a full HC).

Benefits from Premium Catalog Services (PCS) centralization

PCS centralization enabled standardization of processes and offshoring to RBS of repetitive tasks. This led total productivity ratio (vendors/HC) to grow from 4.25 at the end of 2013 to 7.7 at the end of 2016, and offshoring ratio (HC offshore/ Total) from 23% to 83%. Centralization allowed to effectively drive innovation of the program, e.g. Proactive PCS was launched in Q2 2016 and new tasks are now provided to vendors (C-returns, Enhance ASIN content, etc.).

The team has also built processes to track vendor satisfaction and work on its continuous improvement:

- yearly general satisfaction survey (last one sent in May 2016: 87% of vendors satisfied or very satisfied with the service);
- How was My Driving (HMD) survey sent to vendors every time they request a task through Paragon (average EU5 results for 2016 is 88% positive response).

Appendix:

Appendix I – AVS Tenets (unless you know better ones)

- Amazon Vendor Services (AVS) is the best deal for eligible vendors: it is a paid service that helps them deliver the best customer experience and maximize their growth on Amazon.
- Amazon Vendor Services (AVS) is restricted to profitable vendors that can grow and large vendors that can become more profitable; we will provide more proactive and personalized support to the most relevant and attractive vendors (to be reviewed after eligibility tenets have been signed off).
- We offer a menu of services and select those which best address vendor needs and opportunities for development, and we obsess over their satisfaction with the service.
- We deliver our service through scalable processes and automation while keeping the right level of interaction with vendors and close collaboration with Retail Category Teams
- We will work with global teams to drive global parity and efficiencies in SVS, where possible.
- We maintain tight controllership on vendor satisfaction, headcount, profitability and compliance to guidelines.
- AVS is funded incrementally to any other terms and its operation is highly profitable for Amazon.

Appendix II – Framework for structuring AVS Brand Specialists teams

A team is defined as a group of Brand Specialists reporting into an AVS Manager exclusively accountable for service quality and productivity, and detached from day to day vendor activity as POC. These teams will have sub teams managed by L5 Brand Specialists and mirroring the retail organization (typically 1 L5 POC per Category leader). These teams may also have experts working across vendors/categories, typically on advanced operational efficiency and traffic activities, and either reporting to sub-teams or to the AVS manager.

<u>Teams should cover at least 40 vendors to be able to manage business variability & keep experts fully utilized</u>

- Teams need to have enough vendors that:
- Vendor growth forecasts are reliable and can be used for HC planning
- Unplanned absences (sick leaves, recruiting voids) do not cause too much stress on the rest of the team
- Operational excellence & traffic experts on the team are fully utilized Based on the below analysis, the minimum team size should have 40 vendors, equivalent to about 15 HC today (assuming GL work is done outside the AVS teams) and 9 in 2018 (when we double productivity).

The below table is based on real 2016 data, adjusted for 2017 growth forecasts. It takes all vendors with an SVS contract on March 31st , divides them into random groups, and looks at how many vendors were in that same group 3 months earlier. While the average group always grows by 11%, small groups of vendors are clearly less predictable than big groups. At 40 vendors per group, 90% of the groups grow at less than 2x the average - the maximum variability that can be managed effectively (standard deviation below 50% of the mean).

Simulation	120 vendors per group	60 vendors per group	40 vendors per group	30 vendors per group	20 vendors per group	10 vendors per group
Total number of vendor groups	10	20	30	40	60	120
Average vendor growth in Q1	11%	11%	11%	11%	11%	11%
Minimum vendor growth in Q1	5%	2%	2%	1%	0%	-2%
Maximum vendor growth in Q1	14%	22%	24%	25%	42%	78%
Standard deviation as % of mean	27%	41%	45%	64%	82%	136%

Teams should have no more than 30 HC to remain manageable

- Effective management guidelines recommend up to 5 direct reports per people manager (note: the average retail L5/6 People Manager has 3.5 direct reports and the top quartile has 5.6 direct reports, which is consistent with our direction).
- L6 AVS managers can have 2 levels of organization (i.e. L4 Brand Specialists reporting into L5 AVS Team Leads reporting into them) below them
- L5 Team Leads will also own vendors, so should ideally manage up to 4 Brand Specialists rather than 5, and half as many vendors as them. 5 directs would be a maximum. As a result, L6 AVS Managers would have up to 5 L5 Team leads + 20-25 L4 Brand Specialists = 25-30 HC per team (25 being optimal).

Divisions with multiple teams should designate SMEs to optimize the ease of driving process standardization

While the AVS Managers will report into retail, it is essential that they collaborate closely with the central AVS team driving the service roadmap so that they can voice their requirements, contribute to the development of tailored solutions and guarantee full adoption of technology, HOTW and offshore processes. To maximize ownership and ease decision making, there needs to be as few AVS managers as possible in direct contact with the central team. On the other hand we need a minimum of 1 AVS manager per division in contact with the central team to ensure that all business voices are heard. As such, in large divisions which have multiple AVS teams, we will need to designate one of the AVS Managers as SME to the central AVS team.

The below table illustrates the team structure across EU5. We still need to define how to treat the very small divisions (Fresh, Digital, Auto/BISS, Luggage) and we need to refine the Brand Specialist HC excluding GL work as a result of the survey.

Appendix III – AVS organizational model by marketplace

					Current State			Future State					
					# Vendors	Current # Brand Specialists	Brand Specialists to Centralize	L6 AVS Managers	L5 Team Lead Brand Specialists	L4/5 Brand Specialists	# Directs per AVS Manager	Total HC per AVS Manager	Vendors by AVS Manager
DE Ralph Kleber	HARDLINES Jens Uwe Intat	Martin Schueler	Office		12	7	4	1	1	3	3	16	41
			PC		29	17	11		2	9			
		Neil Sentance	Camera		12	4	3	1	1	2	4	19	58
			Wireless		20	8	5		1	4			
			Home Entertainment		10	7	4		1	3			
			Musical Instruments		5	2	1		0	1			
			CE		11	7	4		1	3			
		Wolfgang Eckert	Sports		16	12	6	1	2	4	4	14	35
			Tools		11	7	4		1	3			
			Lawn & Garden		8	7	4		1	3			
		Vinzenz Greger	Home		28	13	7	1	2	5	4	8	35
			Furniture		7	3	2		0	2			
			Major Appliances		11	6	3	1	1	2	3	11	39
			Kitchen		13	8	4		1	3			
			Toys		15	7	4		1	3			
	CONSUMABLES Rocco Braeuniger	David Jackson	Baby		15	6	4	1	1	3	6	16	41
			PCA		10	3	2		0	2			
			Pets		5	5	3		1	2			
		Fiona McDonnell	Wine & Spirit	UK	4	5	4		1	3			
			Wine & Spirit	DE	7	6	4		1	3			
		Matthias Peukert	Beauty		15	14	9	1	2	7	5	21	52
			Food		15	8	5		1	4			
			Prime Pantry		0	1	1			1			
			HPC		22	10	6		2	4			
	Media	Nils Graef	Books		34	4	3	1	1	2	4	14	84
		Eberhard Wirtz	Software		5	1	1		1	4			
			Video Games		13	6	4						
		Christian Hoeweler	DVD		17	6	4		1	3			
			Music		15	5	3		1	2			
UK Doug Gurr	HARDLINES Andrew Milliken	Fergal Gara	PC		26	18	13	1	2	11	2	13	35
			Office		9	0	0		0	0			
		Jamie Heywood	CE		8	17	12	1		9	3	13	33
			Wireless		11	1	1		3	1			
			Home Entertainment		7	0	0			0			
			Camera		7	0	0			0			
		Richard Marriott	Home		27	11	8	1	2	6	4	13	43
			Kitchen		13	6	4		1	3			
			Musical Instruments		3	1	1		0	1			
		Victoria Nelson	Toys		8	5	3	1	1	2	5	11	38
			Sports		13	6	4		1	3			
			Lawn & Garden		7	2	1		0	1			
			Tools		10	3	2		0	2			
	CONSUMABLES Patrick Pondaven	Carlo Mocci	Beauty		14	22	16	1	3	13	3	16	39
			HPC		14								
			PCA		11								
			Baby		10	4	3	1	1	2	5	13	43
			Food		6	3	2		1	1			
			Prime Pantry		12	1	1		0	1			
			Pets		11	5	4		1	3			
			Wine & Spirit		4	5	4		1	3			
	MEDIA Alison Forrestal	Dan Mucha	Books		9	5	3	1	1	2	4	10	24
		Russel Jones	Software		2	1	1		0	1			
			Video Games		4	2	1		0	1			
		Marc Bordier	DVD		7	5	3		1	2			
			Music		2	5	3			3			

Note: "Number of Brand Specialists to centralize" exclude those that will remain in the GL. Figures based on end of Jan 2017 actuals.
Proposal for UK Media is draft. We will deep dive on UK Media Peculiarities and come with a final proposal by mid-May.

			Current State			Future State					
			# Vendors	Current # Brand Specialists	Brand Specialists to Centralize	L6 AVS Managers	L5 Team Lead Brand Specialists	L4/5 Brand Specialists	# Directs per AVS Manager	Total HC per AVS Manager	Vendors by AVS Manager
FR Frédéric Duval	HARDLINES 1 Valery Raymond	Office	8	3	2	1	0	2	4	12	30
		PC	22	13	10		2	8			
		Camera	8	4	3	1	1	2	6	13	31
		CE	5	4	3		0	3			
		Musical Instruments	3	1	1		0	1			
		Wireless	15	9	7		1	6			
	OTHER HARDLINES Céline Vuillequez	Home	10	5	3	1	1	2	4	17	45
		Kitchen	5	3	2			2			
		Sports	7	4	3		1	2			
		Tools	8	5	3		1	2			
		Toys	15	8	5		1	4			
	CONSUMABLES Cyril Dumanois	Baby	9	5	3	1	1	2	4	11	39
		Food	2	1	1		0	1			
		Health & Beauty	22	9	6		1	5			
		Pets	6	2	1		0	1			
	MEDIA Hughues Pitre	Books	4	1	0	1	0	0	3	3	19
		DVD	6	2	1		0	1			
		Music	2	1	0		0	0			
		Video Games	7	3	1		0	1			
SOFTLINES Susan Saideman	Susanne Tebartz	Apparel	10	24	17	1	3	14	4	18	12
		Luggage	2	2	1		0	1			
	John Lohnas	Shoes	45	36	24	1	4	20	4	24	45
	Natali Stojovic	Jewelry & Watches	14	22	15	1	3	12	3	15	14

Note: "Number of Brand Specialists to centralize" exclude those that will remain in the GL. FR figures based on end of Jan 2017 actuals. Softlines figures based on March actuals including BIS HC and counting vendors under Pan EU contract as one.
Note: Softlines will minimize remote reporting of L4s to the extent possible and manage it effectively when needed

			Current State						Future State											
			# Vendors		Current # Brand Specialists		Brand Specialists to Centralize		L6 AVS Managers		L5 Team Lead Brand Specialists		L4/5 Brand Specialists		# Directs per AVS Manager		Total HC per AVS Manager		Vendors by AVS Manager	
			IT	ES	IT	ES	IT	ES	IT	ES	IT	ES	IT	ES	IT	ES	IT	ES	IT	ES
IT & ES François Nuyts	HARDLINES 1 Giacomo Trovato	Camera	7	7	3	3	2	2	1	1	2	1	10	6	2	1	12	7	34	35
		Wireless	10	13	4	3	3	2												
		CE	10	10	6	3	5	2												
		Musical Instruments	6	5	1	0	1	0												
		Office	1	0	0	0	0	0	0	1	1	2	4	8	3		15		54	
		PC	22	30	6	14	5	10												
	OTHER HARDLINES Ruth Diaz	Home	3	7	1	2	1	1	1	1	2	2	10	6	2	2	12	8	37	44
		Kitchen	10	8	5	5	4	3												
		Lawn & Garden	0	3	0	0	0	0												
		Sports	7	13	4	3	3	2												
		Tools	9	2	4	2	3	1												
		Toys	7	10	3	5	2	3												
	CONSUMABLES Stefano Martinelli	Baby	6	5	3	2	2	1	0	1	1	1	5	6	2		13		48	
		Food	1	2	1	2	1	1												
		Health & Beauty	10	24	5	8	3	5												
	MEDIA Giorgio Busnelli	Books	0	0	1	0	1	0	1	0	1	1	6	3	2		11		26	
		DVD	6	5	4	3	3	3												
		Music	3	0	1	0	1	0												
		Video Games	7	5	3	2	2	1												

Note: "Number of Brand Specialists to centralize" exclude those that will remain in the GL. Figures based on end of Q3 2017 estimates
Note: We will minimize remote reporting of L4s to the extent possible

Appendix IV – GL activities performed by Brand Specialist by Division

Preliminary results based on replies of 70% of Brand Specialists

Non-SVS Work by Marketplace and Product Line

	DE	UK	FR	IT	ES	LUX	Total
Hardlines 1	36%	29%	25%	16%	27%	0%	29%
Hardlines 2	48%	30%	34%	29%	44%	9%	39%
Media	35%	43%	51%	21%	14%	0%	37%
Consumables	37%	26%	37%	35%	32%	0%	33%
Softlines	31%	31%	25%	34%	0%	0%	31%
Total	**39%**	**30%**	**32%**	**25%**	**33%**	**9%**	**33%**

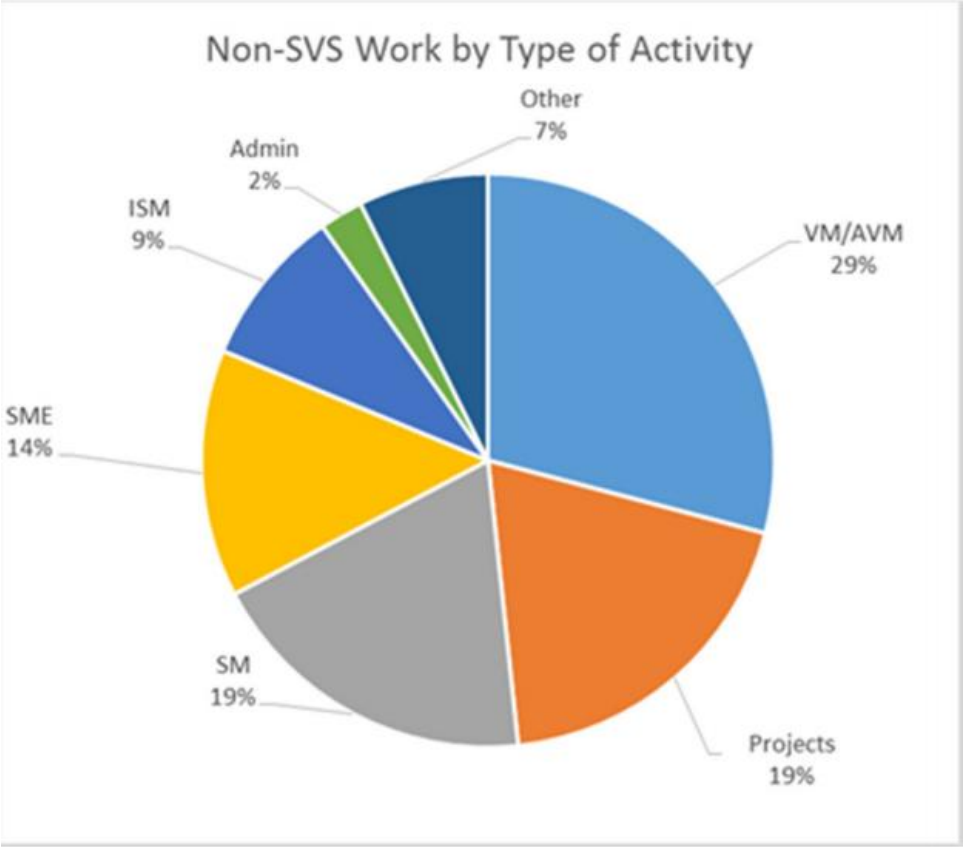
Distribution of Brand Specialists by % of time spent on Non-SVS Work

Percentage of Non SVS Work	% of Brand Specialist
0% to 10%	16%
10% to 20%	17%
20% to 30%	17%
30% to 40%	17%
40% to 50%	13%
50% to 60%	8%
60% to 70%	5%
70% to 80%	3%
80% to 90%	1%
90% to 100%	1%
100%	3%

Non-SVS Work by Tenure

Tenure	% of Non SVS Work
0-6 Months	21%
6-12 Months	28%
12-24 Months	36%
More than 24 Months	48%
Total	**33%**

Non-SVS Work by Type of Activity



Top Non SVS Activities

Activity Type / Activity Name	% of Non-SVS Time
VM/AVM	
Negotiate Vendor terms	2,2%
Identify SIC/SMT and 3P Selection Gap	2,0%
Prepare/Organize/Attend WBRs	1,9%
Organize/Attend othe meetings	1,8%
Build and share reporting	1,8%
SM	
Schedule promotions, run merchandising and push VSS recommendation	1,7%
Build Promotion Plan and setup CCOGS agreements	1,6%
Schedule on Merchandising Campaigns	1,3%
Monitor and report the performance of promotions	1,1%
Monitor sourcing of promotions and promotion ordering	1,0%
Projects	
Projects driving category roadmap (e.g. launch new programs, try before you buy, private label, heavy bulky)	4,7%
Projects to improve category visibility (e.g. gaming event, back to school campaign)	3,2%
Projects to improve productivity (e.g. queries, tools, macros, SOPs)	3,0%
Projects to improve quality of work (e.g. training, wikis, templates, trackers, checklists)	2,7%
Projects to improve quality of category reporting (e.g. dashboards, downstream analysis)	2,0%
SME	
SME/POC for retail topics (e.g. selection, deals, bundles)	3,8%
SME/POC for Amazon productivity initiatives (e.g. HOTW, VSSP)	2,6%
SME/POC for retail processes (e.g. LoS, LBB, MCP, Andon Cords, TT)	1,7%
SME/POC for retail tools (e.g. dashboards, trackers)	1,7%
SME/POC for Amazon cross-category programs (e.g. Private Label, Heavy Bulky)	1,6%
ISM	
Deep dive on availability, create and execute action plan	1,6%
Monitor & investigate OOS	1,4%
Support vendors on implementation of Operational Excellence initiative (EDI, Dropship, We Pay, Direct Import ...)	1,1%
Manage unhealthy inventory and overstock	0,8%
Support resolution of shortage issues and drive vendor improvements	0,8%
Admin	
Attend Trainings / Brownbags / Office Hours	1,5%
Mentor / Coach other employees	0,5%
Organize Trainings / Brownbags / Office Hours	0,2%
Interview candidates for recruitment	0,1%
Manage direct reports	0,1%
Other	**0,0%**
Other Activities outside of your SVS Role	7,5%

Project 3: PCS Solution Enhancements

Introduction

RBSPCSWorkflow is a platform to manage task for PCS. PCS is a premium catalog service that Amazon offers to its vendors. In this service Amazon associates works to improve catalog quality on behalf of vendor. Currently this service is being offered in EU & US region.

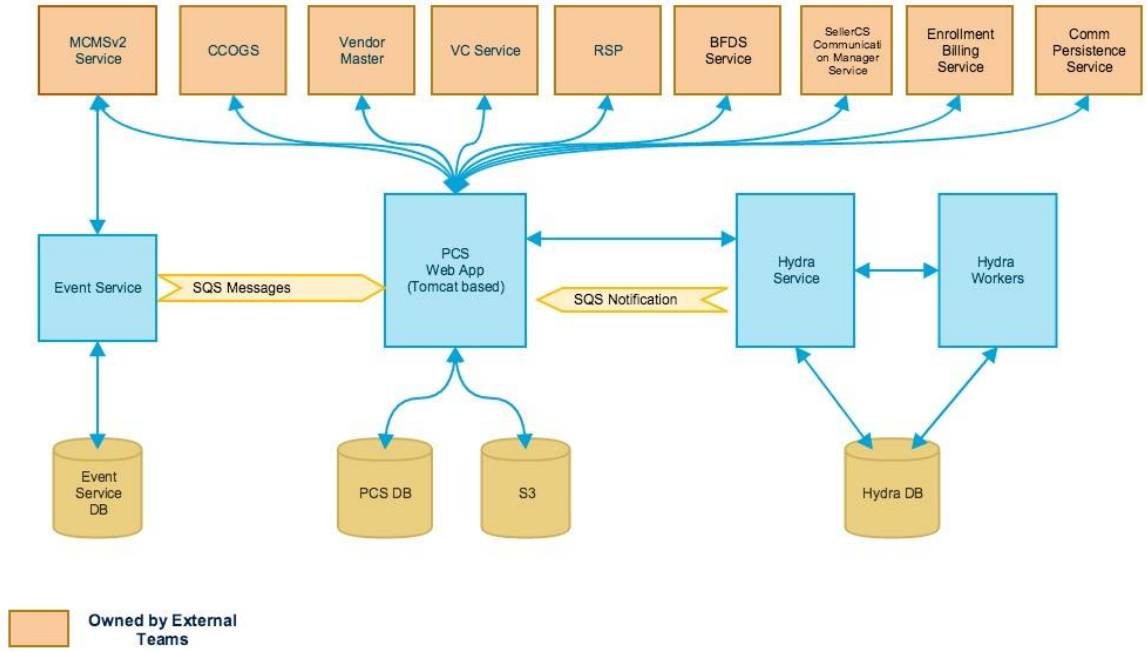
As of now associates manage and track their PCS task on tt and some excel based systems. RBS PCS workflow will eliminate these systems and will provide a single platform to do all activities. This platform will provide better metrics based reporting and trackability.

In R1 (Release1) of PCS workflow, we are targetting to provide platform for Agremeent life cycle management, which will result in 2.4% of productivity gain.

Use Case

- VM or associates should be able to log an agreement. Vendor details like contact name, email etc. should be fetched from VM service automatically. Signature is present in CCOGS agreement should also be checked. VC rights should also be updated automatically.
- VM or associats should be able to view the details of an agreement and should be able to update the information of agreement (only permissible fields).
- Associates should be able to assign agreement to himself/herself or to someother associates so that they can start working on them.
- Associates should able to change the status of an agreement, status like Active, Closed etc.
- VM or associates should be able to search agreements based upon search criteria to maintain reporting.
- VM or associates should not be able to log duplicate agreements.
- Vendor Tracker and WBR should be automatically generated as a part of reporting.

System Components



Dynamo Db

Dynamo DB will be used as data base to store agreement entity. We will be using composite key based secondary GSI to query agreements based on different-different criteria. Alternative of Dynamo DB was to have relational structure and use RDS. No doubt our data is hierarchal but we can also model it in Dynamo DB also to get scalability and avoid data disasters. [As part of rolling stone project, throughout the org we have mandate to build any new application on our in-house dynamo]

Sample Json representation of Pcs Agreement is like below (All fields are not mentioned here):

{

"CONTRACT_ID": "SVS",

"CREATE_DATE": 1455807812580,

"MARKET_PLACE": "FR",

"PCS_AGREEMENT_ID": "PCS.SEBF3",

"USER_LOGIN": "placzekb",

"BRAND_CODE": "SEBF3",

"STATUS": "active",

"ASSIGNEE": "abcd",

"version": 1 cont.....

}

Search on based on primary index i.e. PCS_AGREEMENT_ID is the default funcationlity offer by dynamo. To accomplish efficient seacrch based on composite business criteria we will be building Secondary GSI with appropiate range key over the table. E.g. If we have to find all active agreements in FR between some date range then we will make a composite key MARKET_PLACE+STATUS and range key will be CREATE_DATE. By this data will be paritioned as per composite key and will be sorted in partions based on range key.

Dynamo db has limitations for having only 5 GSI, we will develop the GSI for most frequently used composite criteria of search. Beyond this all queries will be scanning the table. To optimize it further we will be using filter functions of dynamo db v2 so that data set will be filtered in dynamo server only this wil help in reducing the network latency.

RSP engine

RSP engine is being used to manage the Agreement life cycle. Other options for workflow engine could have been Herd or AWS SWF but introducing these new tech stack will become the overhead for us. RSP internally uses the SWF and it is being used in several RBS Workflow for long.

Vendor Master

VM service is being used to get the vendor data from vendor code.

Contra Cogs

CCOGS service is being used to verify the ccogs agreement supplied by VM or associate.

VC Services

VC service is being used to update the vendor rights automatically in vendor central using VSSVendorService

Steps (Algorithm) involved in this process are:

- GetVendorGroup: In this we call VSSVendorService to get list vendor group using *GetVendorGroupsBySearchValues*to get list of vendor groups for a vendor code. Then, we select vendor group based on following criteria:
    - If List of vendor group contains only one vendor group then return that vendor group.
    - If List of vendor Group contains more than one vendor groups then we compare them using business group and creation time of vendor group

        - Use*GetAllBusinessGroup* to get Map of BusinessGroupId and BusinessGroups for given marketplace then find businessGroupId for input product group then find out which all vendor groups have this business group as their primaryBusinessGroup
        - If filtered list has only one vendor group, then return that vendor group else most recently created vendor group
- UpdateVendorCentralRights: Once you got the Vendor Group then get vendor group right hierarchy using *GetVendorGroupRightHierarchy*
    - If List<Right> contains some rights then create List<RightChangeData>and append it with new right needs to be added. Else create List<RightChangeData>with just new right Ids.
    - Call *ChangeVendorGroupOverrides*api to override rights for given vendor group with new List<RightChangeData>

Dev Guide for APIs:

- *GetVendorGroupsBySearchValues:*
    - Input:
        - VendorGroupSearchParameters: vendorCode
        - GetVendorGroupsBySearchValues.Reques: MarketplaceId (Obfuscated)
    - Output:
        - List<VendorGroup>
- *GetAllBusinessGroup*
    - Input:
        - GetBusinessGroups.Request:
            - Marketplace (e.g “US”,”UK”)
            - MarketscopeId (e.g “US”, “EU” – for all EU marketplaces)
    - Output:
        - Map of business group id and business group: Map<String, BusinessGroupList>
- *GetVendorGroupRightHierarchy*
    - *Input*
        - GetVendorGroupRightHierarchy.Request: vendorGroupId
    - Output
        - List<Right>
- *ChangeVendorGroupOverrides*
    - *Input*
        - List<RightChangeData> - with new rights appended keeping this order (67561, 67741, 67431, 81691, 67441)
    - *Output*
        - Void Assuming no exception as success.

Tomcat Based Web UI

This will be Tomcat web app based on MVC using Spring & Struts. Another alternative for UI stack was to develop on RSP itself. RSP based UI is in DOJO and takes time to develop and to extend currently present RSP widgets that's why we are here opting Tomcat based standalone solution. This way we will speed up the development and can use good UI stack. It will also remove one dependency from RSP.

Operations Performed by WebApp

- It will take care of storing all the related information of PCS into dynamo DB using a DAO layer.
- It creates various workflow executions using RSP integration
- Integrates with various services and gets relevant information using the APIs

DataWare House (DW)+DJS

DW will used as data strore for reporting purpose. Data will be pused into using DCS feed on daily basis. Daily jobs will be invoked by DJS. DJS will be used as job scheduler as it provides better job management, can run job on single host across cluster and provide alarm setup easily. Instead of DJS host based cron could have been used but to maintain failure and to run on single instance across cluster could have been a tedious job to do. AWS Lamba was also one of the option but it can't invoke apollo service.

PCS platform

Jobs and Items Workflow Platform

Problem Statement

Currently we have developed a Premium Catalog Service which is used by the RBS associates to improve the quality of catalogue data for the vendor. There are 14 different types of activities or jobs which are done by the associates on each ASIN (referred to as Item in PCS world). Few examples include - Title correction, Image upload, Browse node correction, etc. The problem is to build a solution that satisfies the following requirements:

- **Generic Job Workflow:** The associates would prefer to track the activities of batch of ASINs and not on per ASIN level. Hence we need to design a generic workflow (job workflow) to enable the associates to track the activities on the batch of ASINs.
- **Generic Item Workflows:** Though the tracking of batch of ASINs is necessary, the actual activities are performed at the ASIN level. Hence we need to have Automated workflows at the ASIN level (Item workflows)
- **Interface between Job & Item workflows:** The job workflow should spawn multiple Item workflows (one for each ASIN) and trigger the Automation at

the ASIN level. Hence the interface between the Job workflows and multiple Item workflows needs to be well defined)

Generic Jobs Workflow

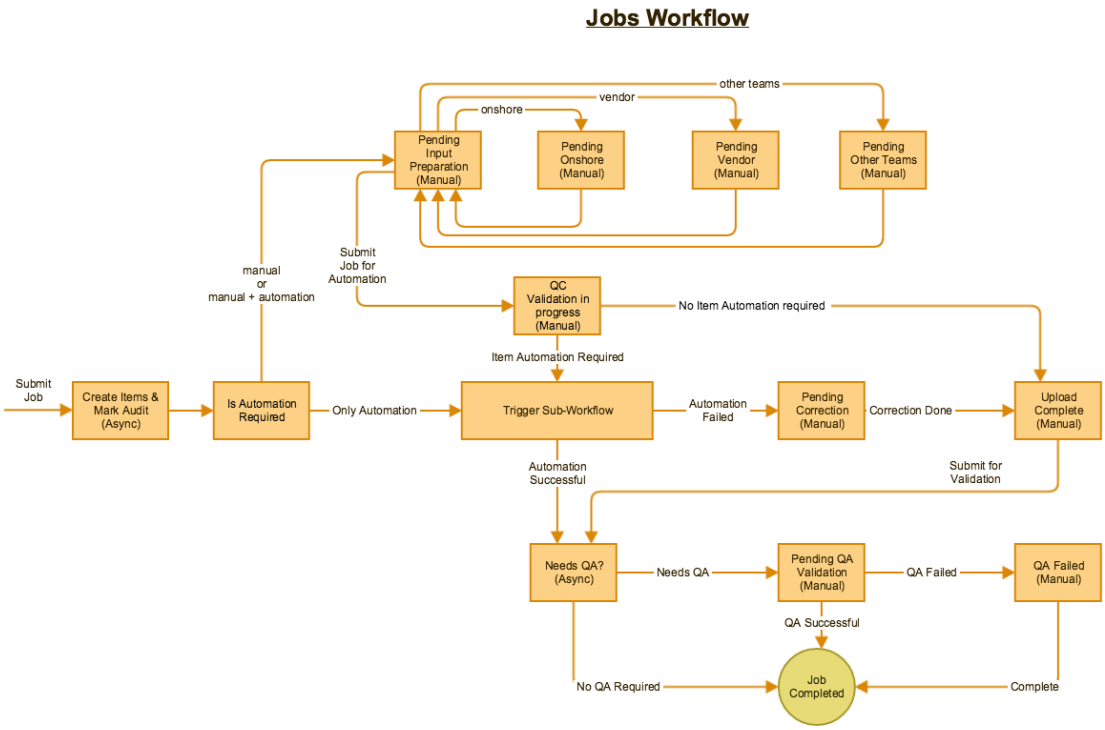
Job Workflow RPDL

Below diagram represents a generic design of jobs workflow. It takes into consideration the workflows of the following types listed below

- Workflows which has only manual steps
- Workflows which has only Automation steps (no manual intervention)

However, it can be easily extended to support workflows which has a mix of automation steps & Manual steps

Based on the type of the workflow, different paths are taken by them as shown in the RPDL below. The type is decided by the context key "jobsWorkflowType".



Starting the Job workflow from UI

- The job workflow is started as soon as a "Create Job" request is placed on the UI. The user is expected to upload a list of ASINs in form of an excel file while placing request to create a new job. This ASIN list is uploaded and stored in a remote S3 bucket.
- The file S3 key would be part of the job entity's ASIN_LIST_FILE_LOCATION attribute in the DynamoDB.
- Job workflow now gets kicked off

ASIN List file is stored in S3 in following hierarchy:
jobId/ASINListFile/filename__timestamp

Similarly, any additional files uploaded by the user during job creation is stored in S3 Bucket in following hierarchy: jobId/AdditionalFiles/filename__timestamp

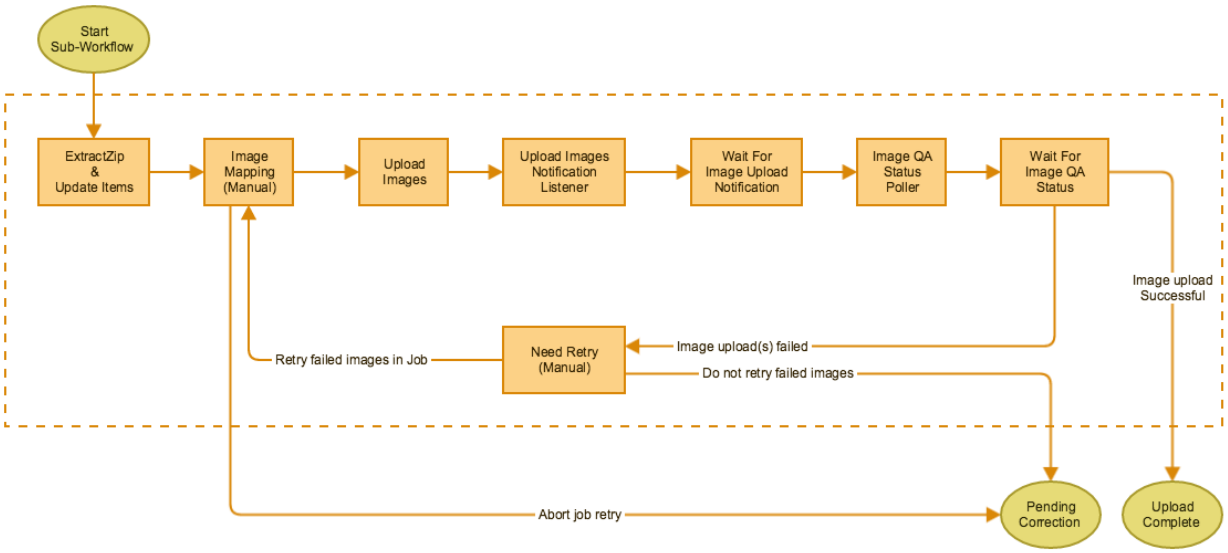
Job Workflow Steps

Start Async step - Create Item Entities & Mark for Audit

- The first Async step in the Job workflow would simply pull the asin list file from S3 and create the Item Entity for each asin and persist it in the DB
- It updates the job with live status (live status field of job says weather the job failed or succeeded. This field could also be updated to indicate any partial failures in any async step)
- Based on the type of job it routes the workflow to manual or automatic path.
- The Items are also marker for Audit randomly based on probability. If there are any items selected for QA Audit, then even the Job is marked for QA Audit

Retry for Image Automation Sub-Workflow

Approach 1: Modelling the Retry with just Job & Items



Task Breakdown

- New History table - This will contain the history of the image upload for the previous try which may be required for any analytics in future.
- RPDL changes - Add a new manual step named - Needs Retry as shown in image upload sub-workflow & modify some transitions to enable the loop
- UI for the new manual step with a "Retry Job" Option
- Clicking on "Retry Job" should save the previous automation try to history + clean up the job sub-workflow context + add a new retry flag to the context

- Image Mapping manual step will need changes w.r.t UI.
    - UI must understand the retry flag in context and must display the images as upload failed/successful
    - Disable the images which are successful from being modified + provision to modify only the failed images
    - On completion the controller should persist the modified images as retryable in the DB
    - UI should have an option to abort the retry (since user may inspect the images which have failed and decide not to retry at this point)
- Upload images async step should identify the retry flag to trigger upload of only selected images
- No changes are forseen as of now in any other async steps in the sub-workflow

Approach 2: Modelling image as Automation Entity

Task Breakdown

- Introduce new Table named Entity with attributes (Entity Id, Entity Type, Entity Details, Entity History, isRetryable, itemId, JobId). With a primary key (EntityId) & GSI (JobId + isRetryable)
- DAO for query, create, update, get
- New blob in Job which will have the entity list on which the workflow will execute. Relavant DAO changes in update/get Job APIs. If this list is empty - this means work on all the entitites associated with the job.
- Refactor all the image upload Async steps to use the new Entity table instead of item image details
- Tasks #2, #3, #4, #5 in the previous approach will also be required

PCS Platform - Support for Debugging

Debug view for PCS Web Application

The idea here is to create debug view for Agreement, communication, Item & Job view pages. This feature will be enabled with a http request param - isDebug=true. On passing a query paramisDebug=true in the Item/Job view URL, the respective page should display all the debugging related information possible which enables the developer to debug the issue as quickly as possible without going to multiple systems like DynamoDB, Hydra, RSP, etc.

Advantages of having a debug view:

Edit

- Most of the issues can be debugged without having to take a peek into the application logs
- No need of being aware of which systems have what information because all the information is now available under one umbrella

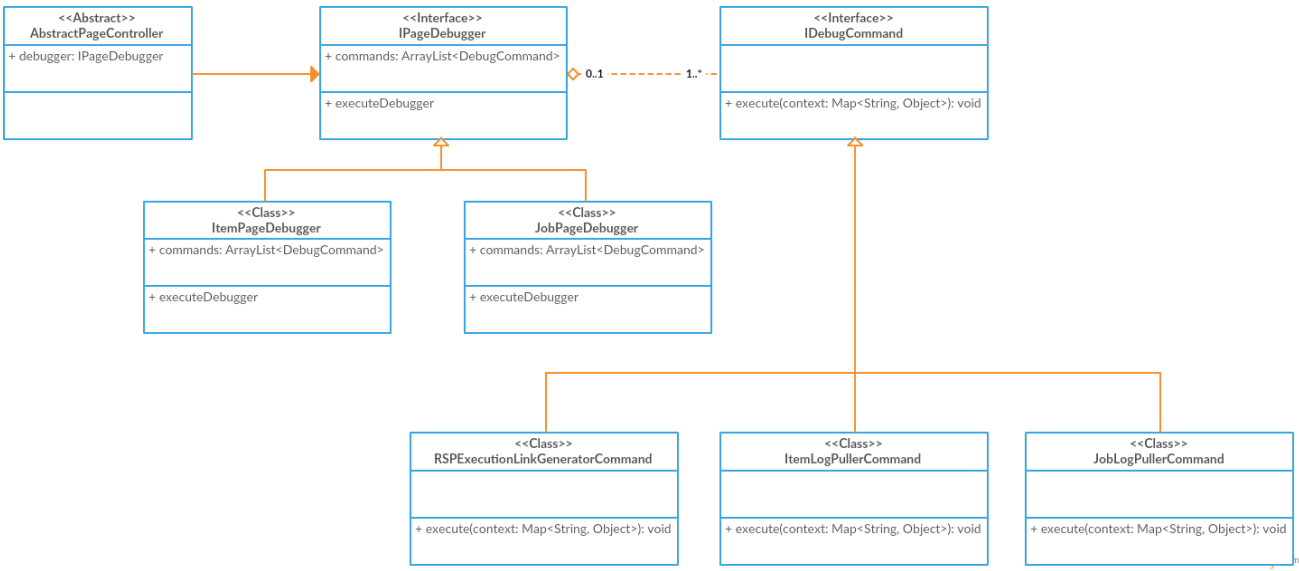
- Makes the life of oncalls and new developers easier initially when they have limited knowledge of the system
- Makes it easier for developers to debug the issues during development without going into multiple systems
- If the view is improved, then even the users will be able to use it to give more details about the issue which they are reporting

Low Level Design

Edit

Diagram below shows the proposed architecture for enabling debugging support on PCS web pages across the application. At a high level, the architecture is based on Command Design pattern and uses number of small commands which can be re-used across multiple pages. These commands can be plugged and played as required to enable different features on multiple pages.

- Write an interceptor/filter which on receiving the isDebug flag
- This interceptor/filter then invokes a command chain based in the page which is being rendered.
- The commands execute and fetch the information from various sources and delegate it to the view (jsp)
- The JSP can simply display it



Job View Page
Currently the job debug view (for only image automation) shows the following informations:

1. Live status of the workflow execution (from dynamoDB)
2. Images uploaded along with the status and failure message, the asin and the variant against which it is uploaded
3. QA status of each image with failure message (if any)
4. Files downloaded from the external sources, status and failure message
5. Automation status of all the automations delegated to hydra, the response from hydra and the failure message if there are any failures

Plan is to also include commands to fetch information from:

6. Hydra - to get the hydra worklogs for each automation using the get automation status API

Extending it to other view pages
With the framework in place, it is very simple to have a debug view for any page in PCS web app. The developer simply needs to write a command class to fetch the information from any source and include it in the chain for the respective view page. The information will be then visible on the respective view page

Admin Update Page

Problem Statement
While creating the Agreement, many fields are filled by user, out of which some fields are non-editable on Update Page.

PCS Dev Team was constantly being approached from Business Team for changing and correcting non-editable fields in DB. Some of such requests are mentioned below:

- https://issues.amazon.com/issues/RBSTech4937 : 47 Requests*
- https://issues.amazon.com/issues/RBSTech5042 : 106 Requests*
- https://issues.amazon.com/issues/RBSTech4830 : 29 Requests*

*- Each request further involved various fields to be changed.

The requests were made due to inability of business team to make the changes in some restricted or non-editable fields, like Submission Type, Product Groups, Service Start/End Date, etc..

Of all the fields, the effort required to change the Dates in DB was the most as for changing a Date, the following flow must be followed:

Generate Date in millis -> Copy and Paste in DB ->Verify in Workflow

No. of Date Changing requests made in above requests:

- https://issues.amazon.com/issues/RBSTech4937 : 21 Requests

- https://issues.amazon.com/issues/RBSTech5042 : 04 Requests
- https://issues.amazon.com/issues/RBSTech4830 : 15 Requests

Due to continuous increase in number of such requests, we came up with the idea to provide Admin Update Page to Limited Users.

Solution Approach

Edit

The admin update page is accessible to only a few users (will be mentioned Admins further).

Following Approaches were discussed to authenticate an admin :

- Provide a secret key
  - Pros :
    - Easy to deploy
  - Cons :
    - May spread to all PCS Users with time

- Provide Dynamic secret key
  - Pros :
    - Admin can generate a key which can be thrashed after usage
    - Unique for every session
  - Cons :
    - Complex Implementation
    - Extra effort required by Admin to generate secret keys per session

- Verify using an LDAP Group
  - Pros :
    - Easy to deploy
    - Authentication automatically done
  - Cons :
    - Requires some planning to maintain the LDAP group, User has to reach to LDAP owner to become Admin

Finally, the authentication of an Admin is done through LDAP verification.

If user is present in the admin LDAP group, he/she is automatically eligible for Admin privilege.

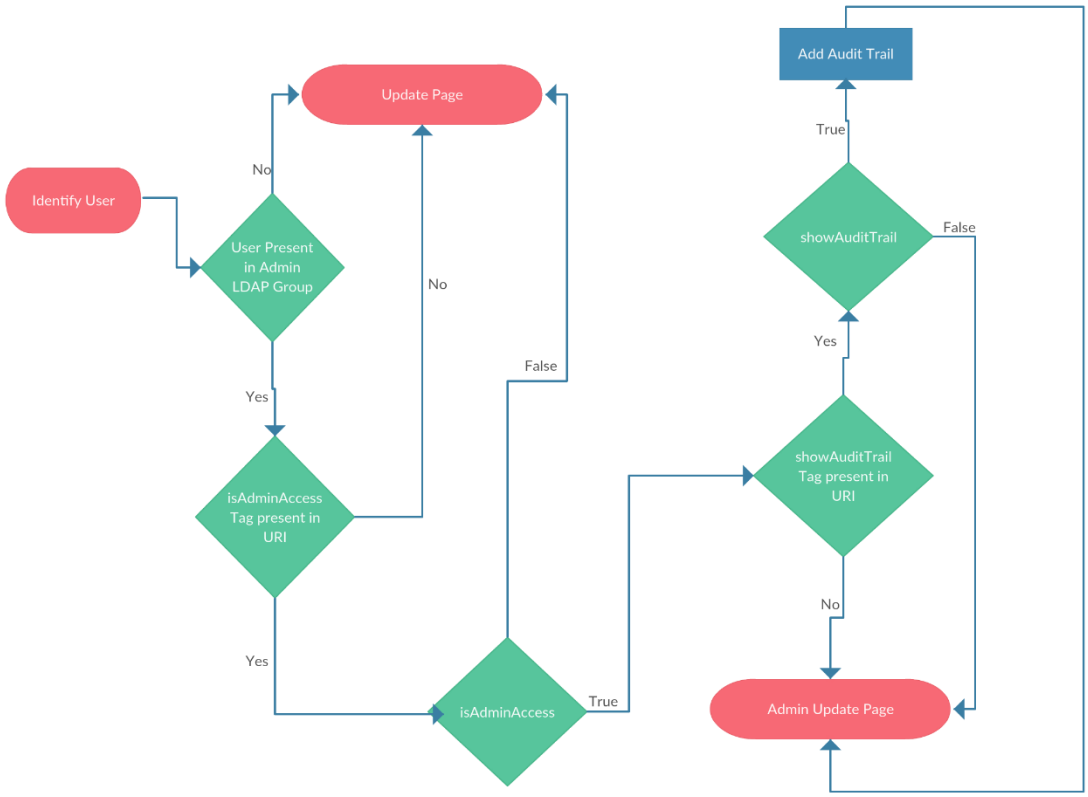
Audit Trail:

This is the special feature added to show an Admin, the trail of the action of Admins on restricted fields.

It is important to record the action on restricted fields by Admin, when we can have multiple admins. Every action by admin on privileged fields is recorded in DB. Admin

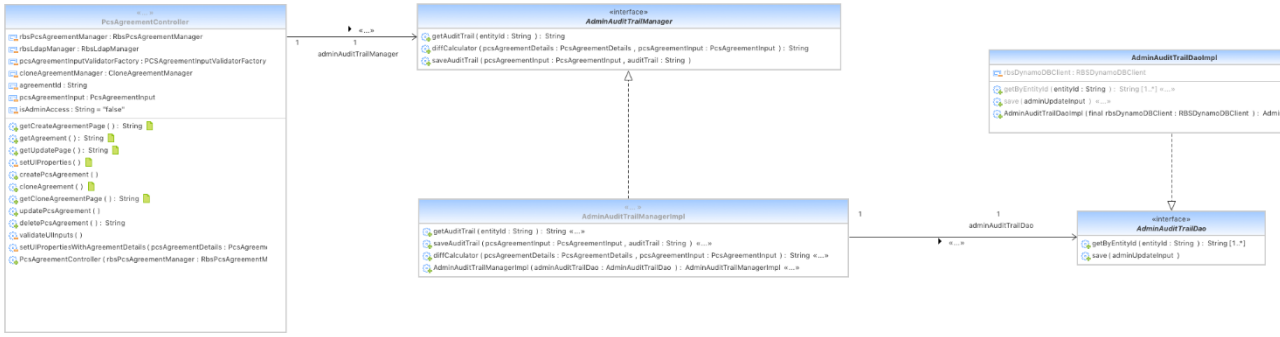
can request this recorded trail to be shown on Admin page by adding showAuditTrail flag in the URI.

Workflow



After visiting Admin Update Page, admin updates the fields restricted to other users and sends control to update the Entity. If the Admin has also updated some restricted fields, the control simultaneously goes to Dynamo DB Table to save the Audit Trail, corresponding to the Entity ID.

Low Level Design



Agreement Update Page for Admin
The above shown flow is followed presently to show the Agreement Update Page.

- An admin goes through two step authentication
- After proper authentication, the Admin is shown the privileged Agreement Update Page
- Restricted fields can be updated here
- Audit trail saved in DB.

Extending Admin Update Logic to other Update pages
With the framework in place, it is very simple to have a Admin Update for other Entities (Communications and Jobs) in PCS web app. The developer simply needs to write a command class to fetch the information from any source and include it in the chain for the respective update page.

Extending the logic for Audit Trail for other Entities
The Audit Trail framework is flexible enough to save and get the Audit trail for any entity.

The developer needs to write separate logic for finding Audit Trail for other entities. To show Admin Page, the developer needs to make changes in the controller. Also after submission, in order to save the Audit Trail, similar changes are to be made in respective function

The DynamoDb Table is designed such that it can be further used for maintaining Audit trail for other PCS Entities as well. The concept of audit trail can even be extended to non-admin users.

Attributes of Dynamo DB Table :

- UUID auditId
- String entityType
- String user
- String entityId
- longtimeStamp
- String log

Audit ID – randomly generated UUID

Entity Type – Ex: Agreement/Job/Communication

Entity Id – ID corresponding to specific Entity. Ex: Agreement ID /Job ID/ Communication ID

Log – Audit Trail

The present Dynamo Db Table structure is so generic that developer can easily use it further to extend some other logic.

Project 4 : Hydra

Problem Statement

We are currently developing a PCS Solution for EU and NA Catalog Services team, which is essentially a platform to automate the PCS process. PCS (i.e. Premium Catalog Service) is a global service spanning UK, DE, FR, ES, IT & US Geographies. The primary objective of PCS is to help the vendors to improve their catalogue quality by doing reactive (vendor initiated) or proactive (Amazon internal initiated) tasks to improve an item related information like title, product description, images, etc. Each task request flowing into the PCS Tool is modelled as a PCS Job. An example for a PCS Job can be:

- Improve the title for 12 ASINS
- Change the image of 89 ASINS
- Correct the browse nodes for 100 ASINS, etc.

Typically each PCS Job involves a set of activities to be done to make the required improvement. These activities may be done by an associate through a single or multiple tools. The process of receiving and completing a job itself has some mandatory steps like getting information, making updates, and doing a QA check on the updates made. A view of this workflow is shown in the wiki here.

Each PCS Job will have one or more ASINs that needs to be fixed and we have modelled these ASINs as a PCS Item. Each PCS Item can go through such a correction process. The final objective of the PCS solution is to automate all these different correction processes that are required to be done under the PCS service. However, the tasks that are done under PCS service are not unique to PCS in the larger RBS org. Other teams in RBS can also benefit from this automation that we build. Additionally, the PCS service and webapp may not be the only client for this automation. We may build other clients that need to submit the request for such automations.

For this purpose, we want to build all these automations as a separate platform such that the automations are completely decoupled from the PCS solution and can be reused by any clients (for example IDQ Team) wanting to leverage the automations. The other key objective is to provide a common framework to log and track these automations that each automation implementation can use instead of writing it on its own. This will result in code reuse as well as consistency on interface for the clients as well.

This wiki talks about the design of Hydra Service, powered by Hydra platform, which decouples the automations completely from the PCS World and the platform on which we are planning to develop all the 14 different types of automations.

Requirements & Objectives

- An automation** could be an activity which needs to be performed on the entire set of items** in the request or it could be an activity which may be

performed by dividing the set of items into smaller subsets (called task) so that the automation can be parallelized into several tasks.

- Build an automation platform which can be a house for several different types of automations.
- An automation could be an activity which takes very less time to complete or might take long time.
- Platform should take care of splitting the item set into smaller subsets and parallelizing the automation whenever possible.
- Platform should be able to scale horizontally by simply adding as many hosts as required.
- Platform should parallelize the automation across all available hosts to exploit the full capacity of all the hosts.
- Platform should be able to handle any burst in the automation requests gracefully without overloading the hosts
- The platform should take care of logging and tracking each automation in a well defined standard way. It should provide necessary metrics out of the box for every automation type.
- There will be several automation types that needs to be supported by the platform. Hence care should be taken to make sure none of the automations starve the others. Like for example set of I/O intensive automations starving CPU intensive automations.

Motivations

There may be multiple clients (like IDQ Team & internal RBS clients) other than PCS who are interested in leveraging the Automations

There could be tenants (internal to RBS Team) who are interested in hosting their automations on this platform so that they can easily scale.

Since we will be hosting several automations, we need to standardize several aspects like logging metrics, handling errors, retries, etc.

Key Concepts

Automation: An automation is an activity which needs to be performed on one or more items. It may or may not be parallelizable by dividing the set of items in the request into smaller subsets.

Request: A request is nothing but the request placed on Hydra for an automation. Each request may or may not consist of set of items.

Task: Task is the actual activity performed on the items present in the request.

The task may be performed on all the items present in the request in one shot . In this case the request will have only one task associated with it.

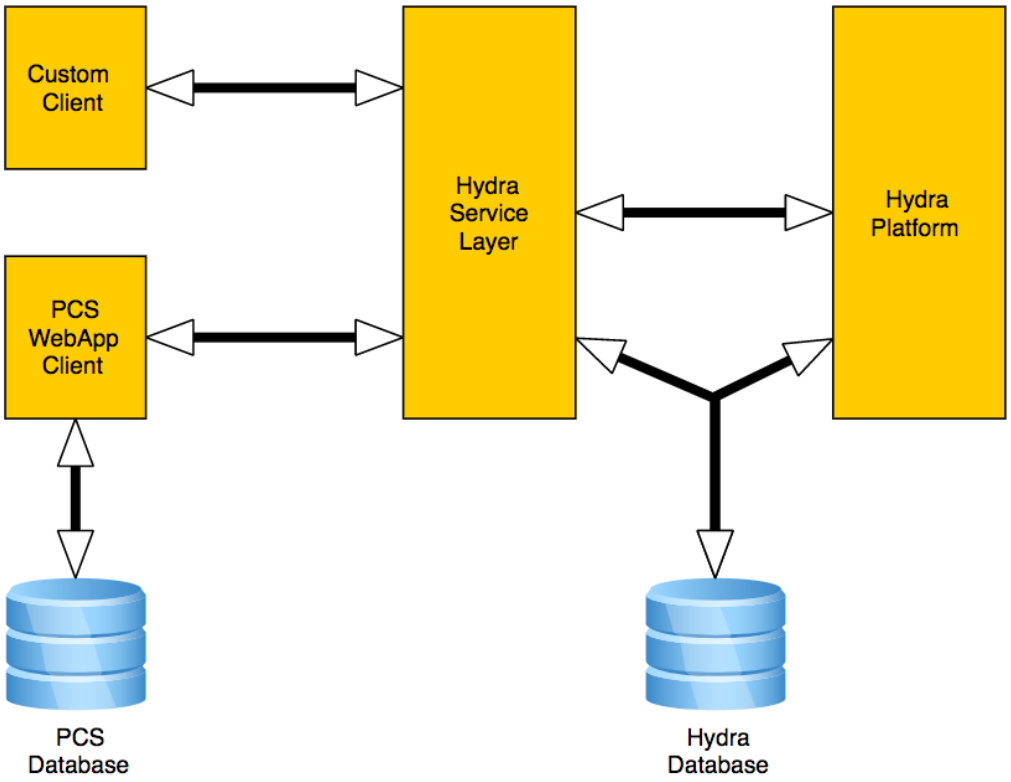
Or, the task may be perfomed on smaller subsets of items, by dividing larger set into smaller subsets, in parallel. In this case the request will have multiple tasks associated with it.

Item: An item is the smallest entity that is identified in the automation request. A task may perform on single item, or a small set of items or all the items at once. Example of an item could be an ASIN, image, etc.

Hydra - High Level Architecture

Today the PCS Application is just a single-tier web application with all the logic written in the web layer itself. Going forward this would be split into Web App layer & the service layer as there would be more than one consumer of PCS Service. The document here however focusses on just the Hydra service & platform shown in the Architecture diagram. This service is expected to handle all the automations going forward starting with the PCS Image upload Automation & Browse Node Automation

.



Hydra Service Layer

Hydra service layer provides an interface for the clients to trigger the automations and get the status of triggered automations.

API for triggering the automations (Common API vs Specific APIs)

There were two options that we evaluated for service APIs (API which would trigger automation) given the number of different automation types we are planning to support.

- Either go with a one common API which supports all the automation types (OR)
- Have one API specific to each automation types

The pros and cons for both the approaches have been discussed in the table below. We decided to go with specific API for triggering different kinds of automation since the kind of input required (especially automation details) is entirely different for different automation types.

API Type	Pros	Cons
Common API	1. Provides a single entry point for all the automation 2. There is no effort of writing a new API each time a new automation type is introduced into the platform. However, a new model specific to the automation type will have to be introduced to enable the client to pass the automation details in the request blob of common API.	1. **Interface with blobs:** The interface is not very clear to the client. The API would have to take a blob/envelop as the input where the automation details specific to different automations can be stitched before placing request. Client would face difficulty in understanding which structure to be stitched into the blob for different automation types. Additionally, we may have to provide a thin client side library to help clients in stitching the different automation details structure into the blob. 2. **Challenges in Validations:** It becomes tricky to handle validations for different automation types as each type of automation expects different types of automation details. 3. **Generic error reporting for all automation types:** Reporting errors throwing automation type specific exceptions would not be possible. We might have to define generic exceptions and a separate hierarchy of exceptions altogether
Automation Specific API	1. **Clear interface to the client:** The client is exposed with a neat and clear interface 2. **Clear way of reporting errors:** We can throw different types of exceptions to report different errors which are specific to the automation types 3. **Clean Validations:** Validation of input for different types of automations are clearly separated out from each other and it becomes clean and manageable. 4. **AAA Access restrictions:** We can restrict the client's access to specific APIs 5. No need of any client library like in case of common API due to absence of blob stuctures.	Each time a new automation type is introduced into the system, developer will have to write a new API altogether. However similar effort is required in defining the model for AutomationDetails in case of common API and the difference in effort can be traded off considering many cons of common API approach.

Common API to get automation status - GetAutomationStatus

We did not consider automation specific API here because all the automation has the same kind of status reporting and it makes very much sense to go with one single API to pull the automation status for any kind of automation which will be implemented on the Hydra platform

Input

Automation Request ID*

Output

The output would be AutomationStatus object with the details of all the automation tasks if only the requestId is passed. However, if taskId is also passed, then the

automationStatusDetailsList would contain only one AutomationStatusDetails object corresponding to the requested taskId.

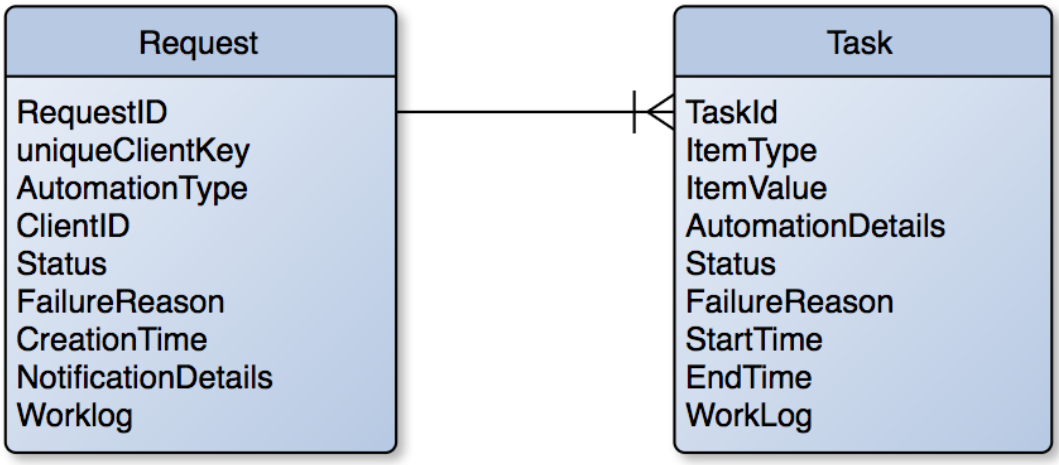
```
class AutomationStatus {
  List<AutomationStatusDetails> automationStatusDetailsList;
}

class AutomationStatusDetails {
  String id;
  String status;
  String failureReason;
  String completionTime;
  List<String> workLog;
}
```

Description

This is a very simple API which the client would use to poll the status of each automation by providing the Request ID and optional Task ID which is returned by the API that triggers the automation.

Hydra Data Layer



Request Entity

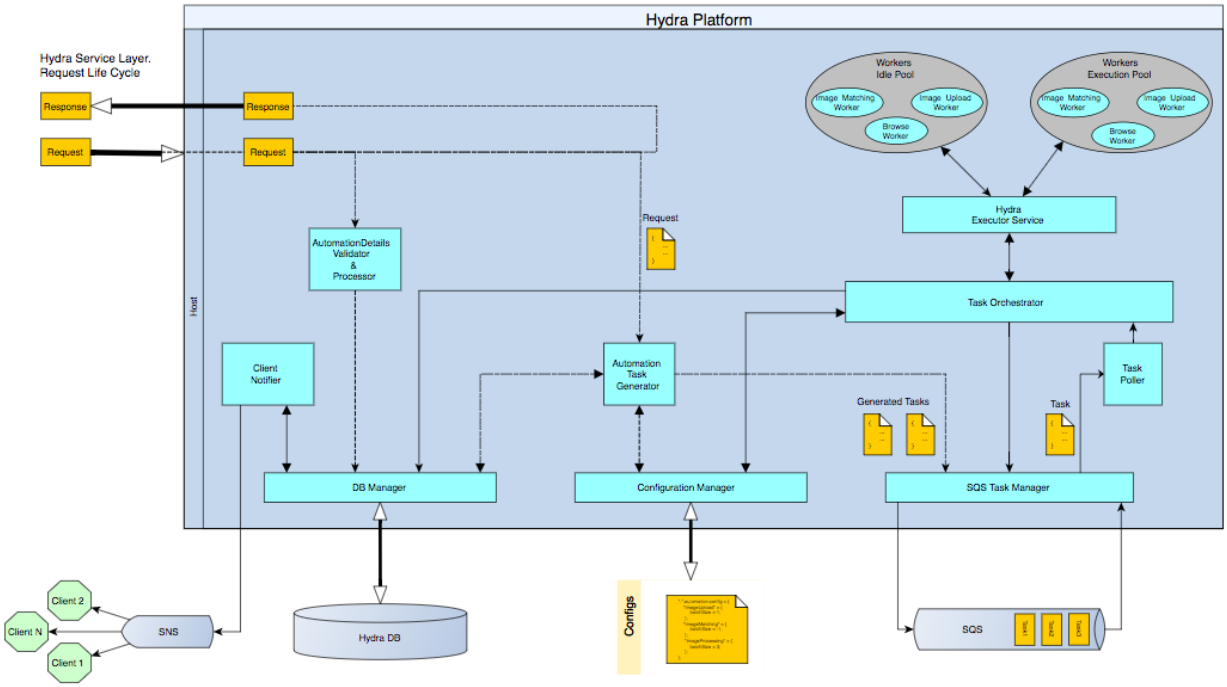
For every automation request that landed on the service, there will be a unique entry in the Request table.

- Request ID - unique id given to the request by the service
- Client Request Key- This key is generated and sent by the client. Hydra uses this key and the corresponding request status to ensure that the API does not create multiple requests in the DB when the client has applied retry logic around the API call. Client should use this id during retries to ensure that multiple retries does not create multiple requests on Hydra platform
- Automation Type - Type of Automation for example - Image Upload, Image Matching, Image Download, Image Processing, Title Correction, Bullet Point Correction, etc.
- Client ID - Client who requested the service
- Status - Overall status of Automation request
- Failure Reason - Reason why the automation failed
- Creation Time - Time corresponding to Request creation
- Worklog - Log of events through which the automation request went through

Task Entity

- Task ID - unique id to one set of items with in the request. If the automation needs to work on all the items in the request, then there will be only one task in the request. However, if the automation can work on smaller subsets of items, then the request can have multiple tasks.
- Item Type - type of item. For example ASIN, Image, ASIN Set, Image Set, etc.
- Item Value - For the itemType = ASIN, the item value would be ASIN. If the itemType = Image, itemValue would be image link and so on.
- Status - Status of the task. Could be IN_PROGRESS, RETRY_IN_PROGRESS, FAILED, SUCCESSFUL, ERROR, etc.
- Creation Time, Start Time & End Time - Time corresponding to Task creation, start & end time of the task
- Automation Details - Meta data required for automation on this batch to complete
- Work Log - Log of events that occurred from creation of task to its completion.
- Failure Reason - Reason why the automation on this batch failed

Components & Flow of Request



Ensuring Request level idempotency

The service ensure request level idempotency i.e. the requests placed into the system will be idempotent even if there are multiple retries. However no idempotency is ensured at the task level. Request level idempotency is ensured as follows:

- A request comes into system with a client key

- A check is made to verify if the request is present in DB and status is IN_PROGRESS.

  - If yes, then simply build the response and send to client

  - If request is present but status is CREATED, then simply delete the request. and proceed to create new request

  - If request is not present, then proceed to create a new request

  Create a new request:

  - create a request in DB with status = CREATED

  - create tasks in DB

  - post tasks into SQS

  - flip the request status to IN_PROGRESS [This means request is now live in the system]

Workers pick the task and check if the associated request status is set to IN_PROGRESS.

- if not, simple delete the task from SQS and do not process the task any further

- If yes, then proceed to process the task

Automation Task Generator (ATG)

ATG is a component which is invoked by the APIs to create automation tasks and queue them for execution. This generator does multiple things:

- Receives the requestId and automation type from API.
- It then quries the Task table using the requestId to figure out the tasks to be generated.
- For each task entry in the dynamoDB table, it queues one task into the SQS.

Task Pollers

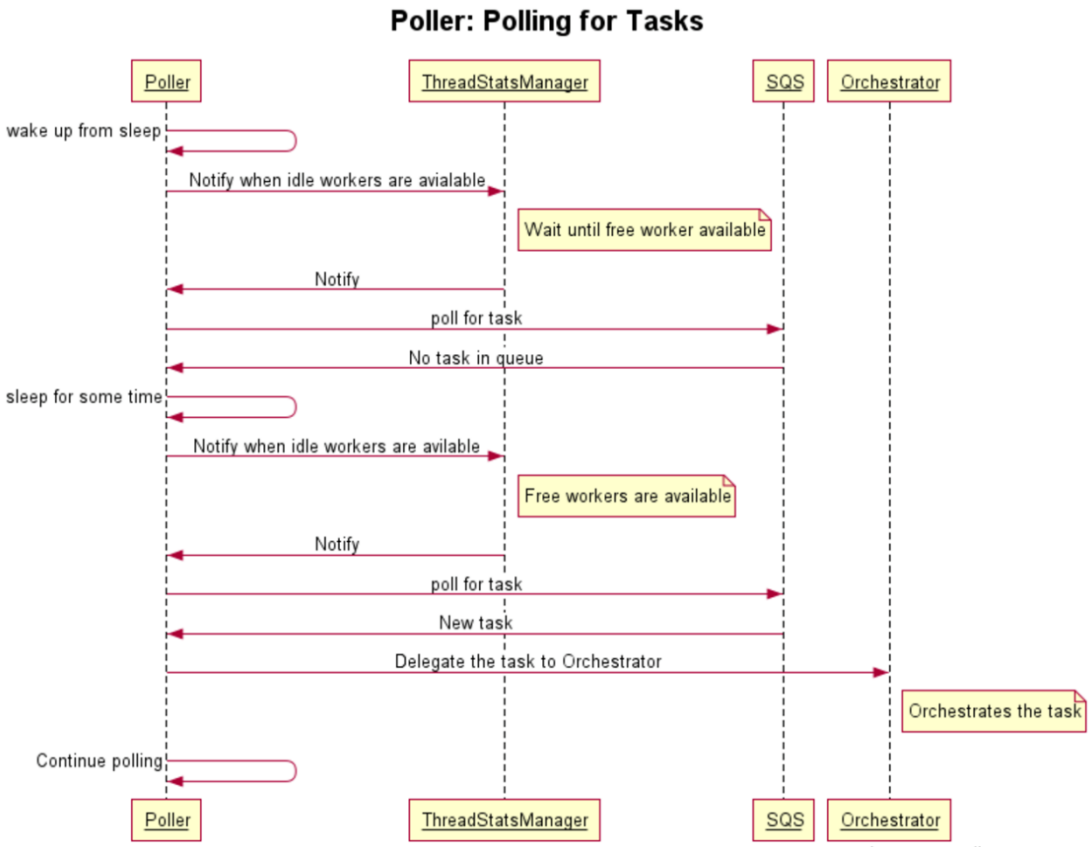
Task pollers keep polling the SQS for new tasks. These poller threads are seperated from the workers. Pollers interact with the Automation Work Allocator to know if there are any free workers before polling the SQS. This optimizes the frequency of polls.

Pros

Having different set of pollers pollng the SQS will reduce the cost compared to all the workers polling the SQS continuosly for tasks because SQS charges for each access request.

Further, pollers interact with the ThreadStatsManager to know the number of free workers before polling the SQS. If there are no workers, then the pollers will not poll the SQS

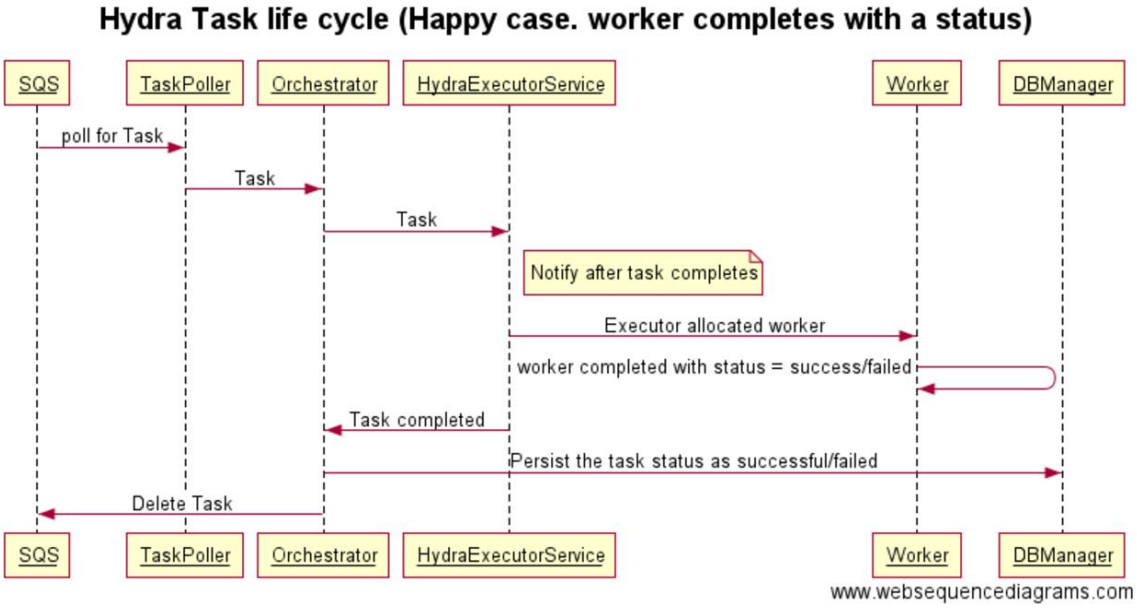
Promotes seperation of concers and keep the workers very simple and loosly coupled to the platform

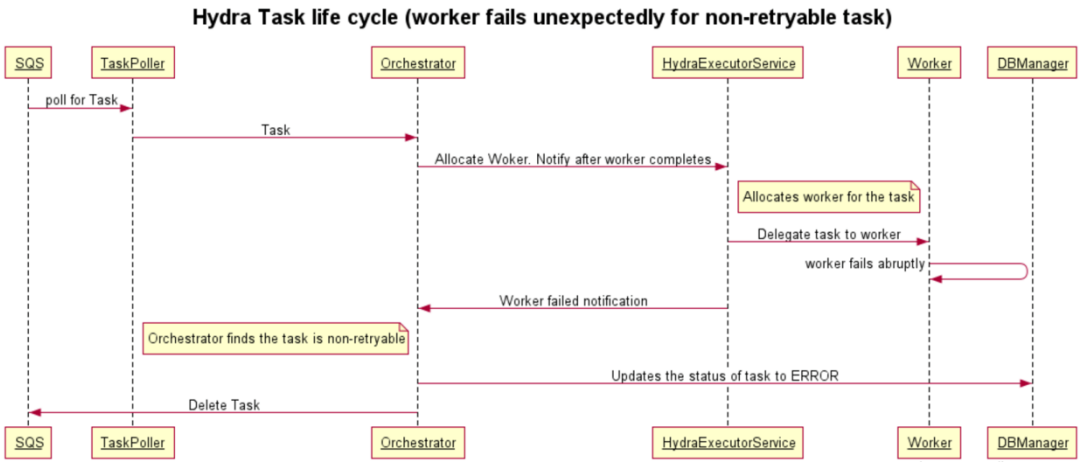


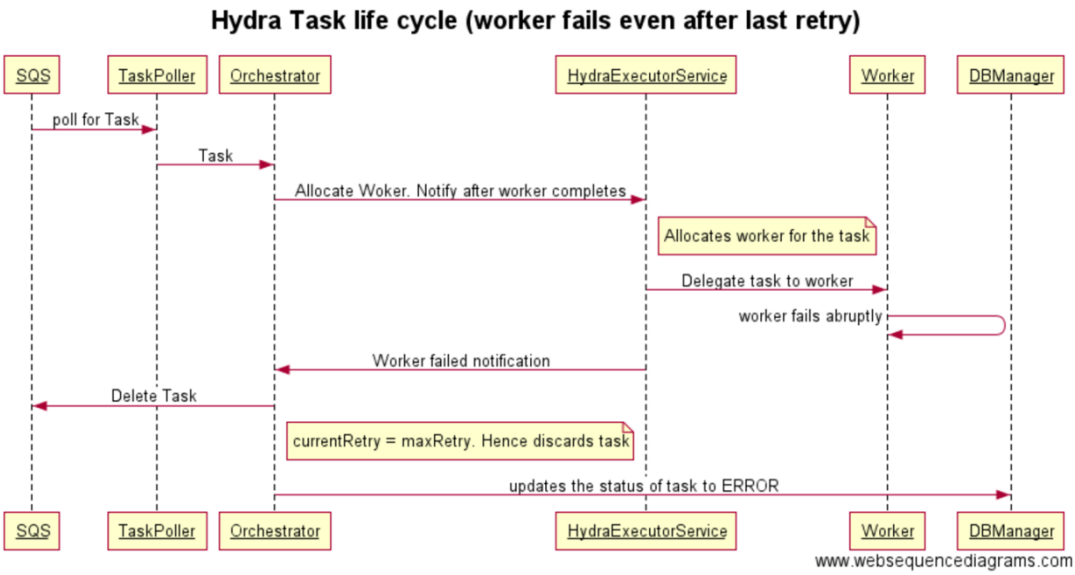
Task Orchestrator

Objectives of Orchestrator includes the following:

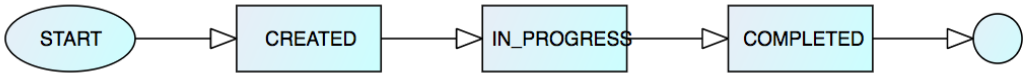
- Receive task from the task poller.
- Delete the corresponding task from the SQS Queue
- Pass it on to the Hydra executor and subscribe for notification from hydra executor about completion of task
- If hydra executor rejects the task, then re-queue the task without retry count being set.
- If the worker completes the task, then update the status for corresponding task in the DB.
- If the worker fails, update status for task as retry_in_progress and then re-queue the task into SQS by incrementing the retryCount attribute of the message.



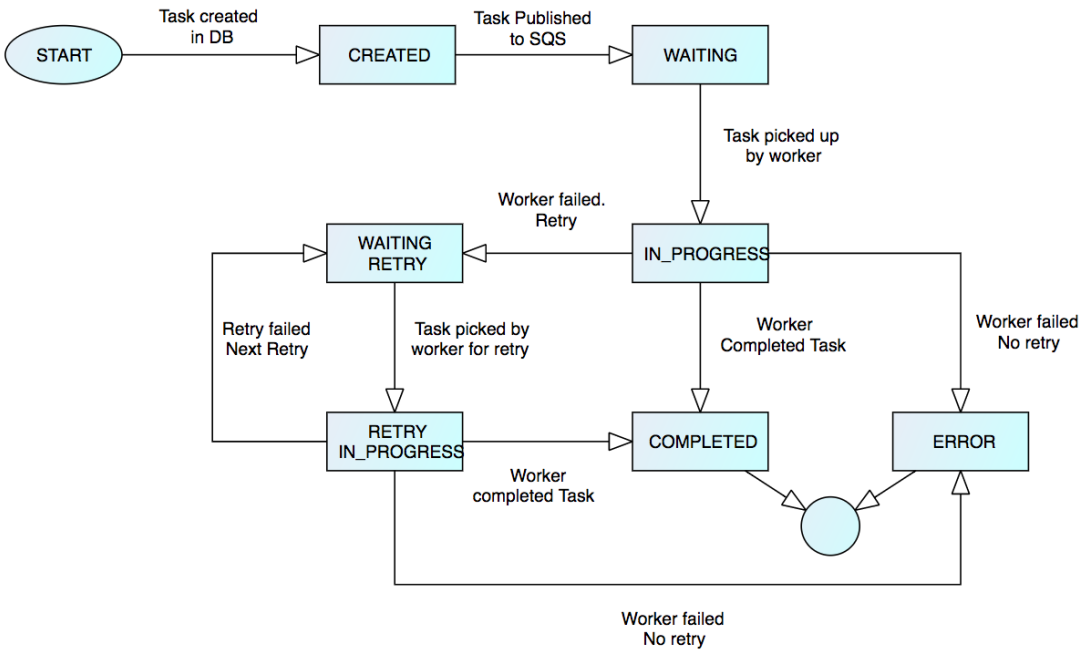




Request life cycle



Task life cycle



Steps to develop a new automation

- [Config Change] Define a new AutomationType
- [Model + Code Change] Define the coral API to be to exposed to the clients which will trigger the automation. Implement basic validations as part of the API
- [Code Change] Implement the Automation Worker
- [Config Change] Configure the pool into which this worker should go in & the number of instances of the worker that can be active on every host
- [Config Change] Configure the parameter required to parallelize the automation by splitting the item list into different batches (subsets)