A Well Mixed Cocktail – A Case Study in Combining Robotic Process Automation (RPA) With Decision Technologies To Improve Case Management

Lloyd Dugan
Lead Business Architect / Solution Architect, Serco Inc. (NA)
lloyd.dugan@serco-na.com

Case Study

General Background

Serco, NA (http://www.serco-na.com/about):

- **Corporate:**
  - Americas division of *Serco Group, plc*
  - Business process outsourcing (BPO) and supporting IT services provider
  - Over 50 years' experience and 10,000 staff in the Americas
  - Multiple government and commercial clients

- **Large BPO and IT support services contract with the Federal Government**
  - Multiple call centers with large number of call center workers doing back-office processing
  - Leveraging Case Management, Decision, and RPA technologies to support service operations
  - PII and IP restrictions on access in Production limit disclosures in the embedded demo

Case Study

Specific Background

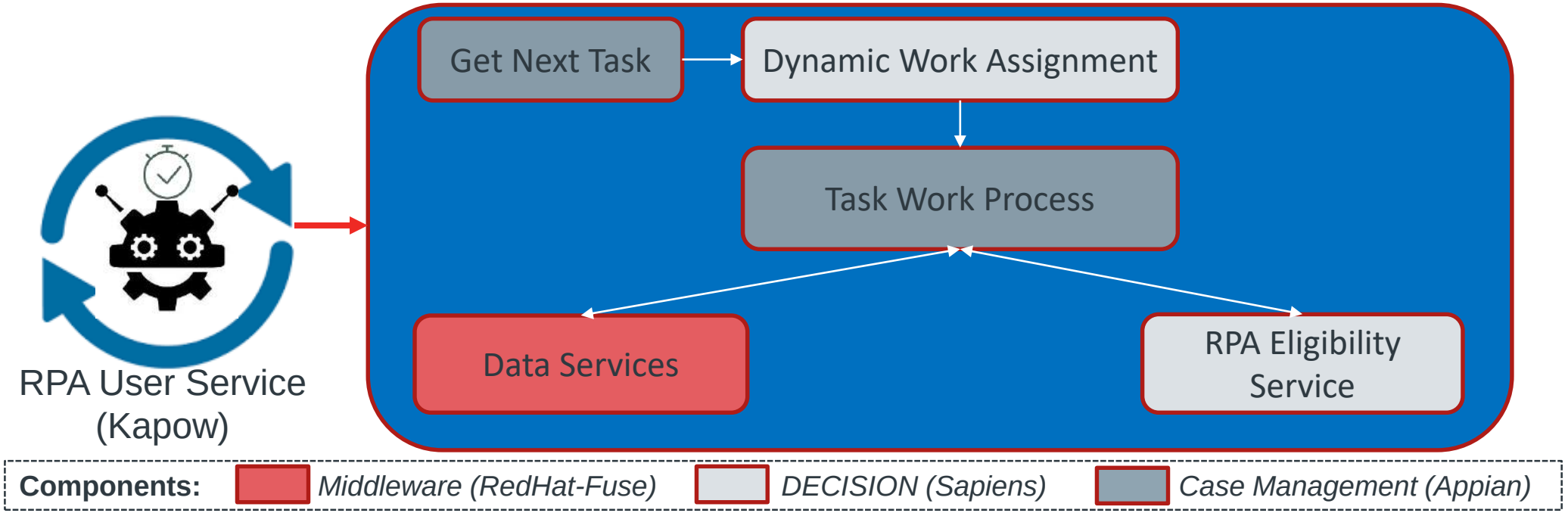
Problem Space and Solution for Assisted Case Work Using RPA:

- **Increasing efficiencies of working tasks in a case management system used by call center workers without sacrificing effectiveness of outcomes**
- **Implement equivalent straight-through-processing (STP) treatment for low-complexity work as way of extending utility of case management solution**

RPA-Enabled Case Management Solution

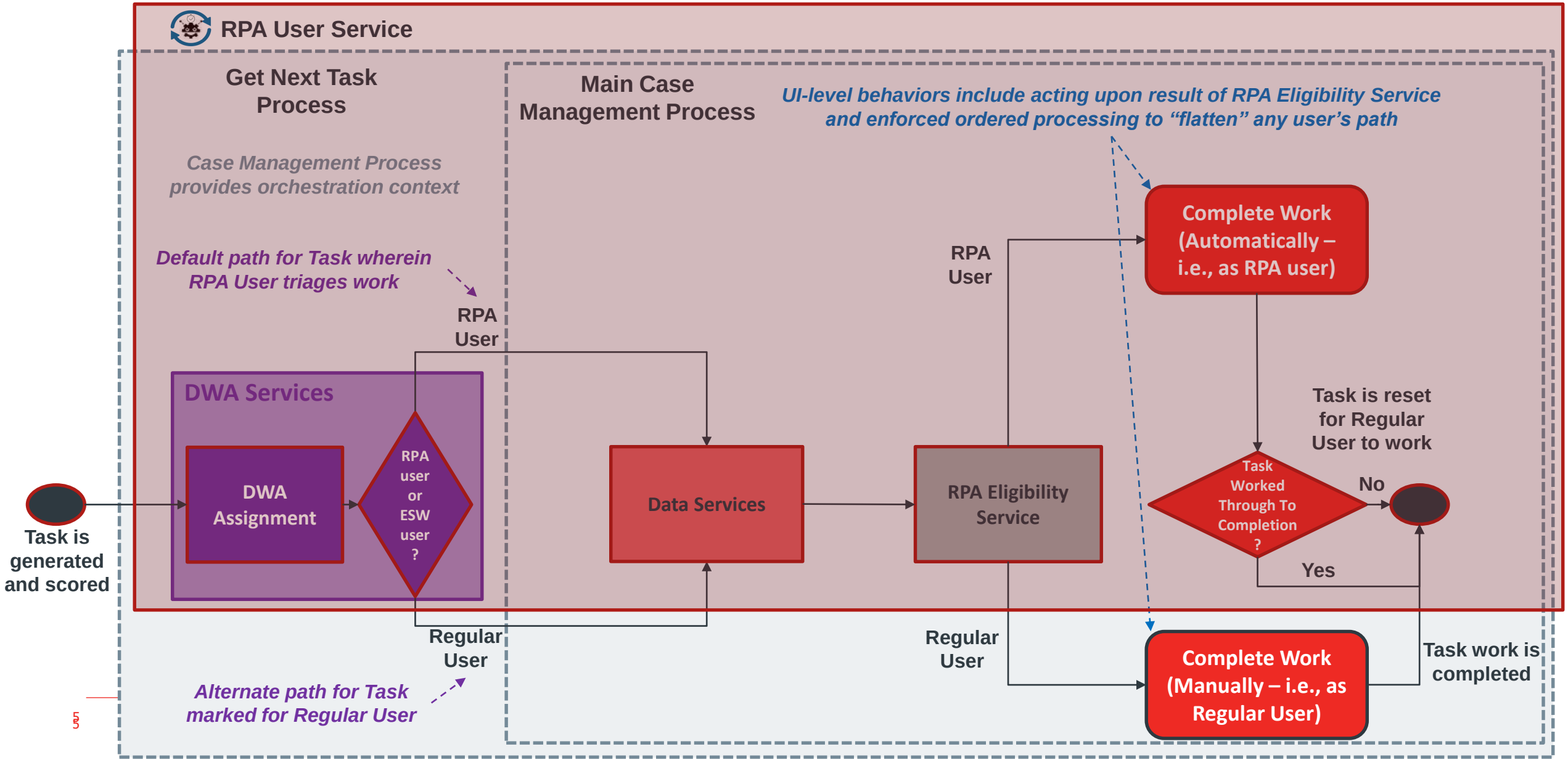
Architectural Overview

- **The RPA Service launches work as RPA Users on a work schedule (as a cron job) like a Regular User.**
- **The RPA Users have appropriate logon credentials and provisioning rights for using the system.**
- **New Decision Service determines if a task qualifies as low-complexity for processing by the RPA User.**



RPA-Enabled Case Management Solution

Functional Flow



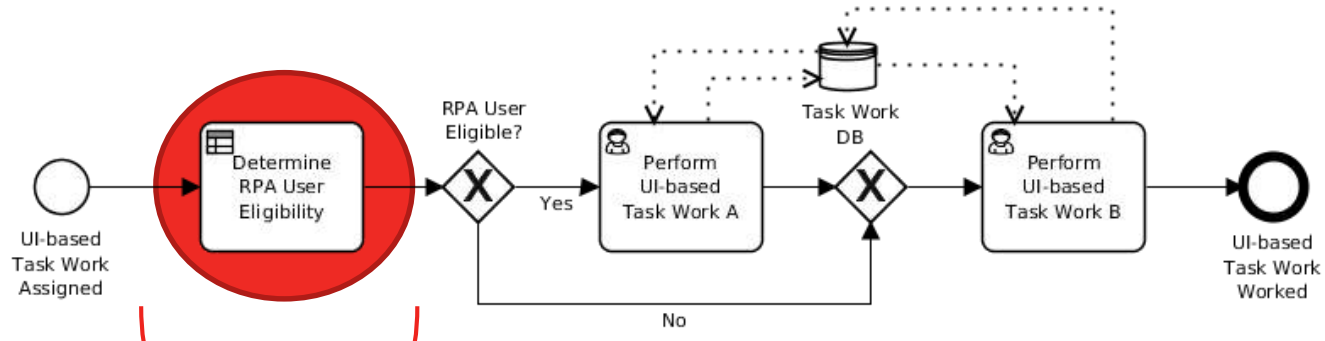
RPA-Enabled Case Management Solution

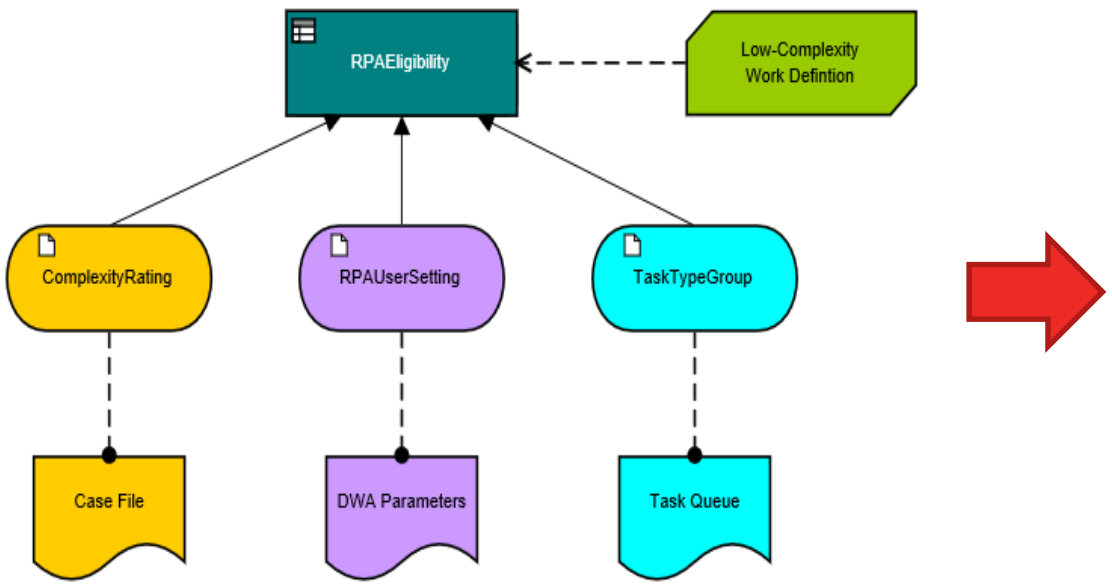
Operational Results

- **Worked approximately 60k tasks over a 15 day period (out of 160+k tasks), avoiding need for Regular Users to work them**
- **Averaged about 12 tasks/minute and took between 4-8 minutes/task at up to 40 RPA User sessions (across 10 RPA User accounts)**
- **Was THE key to meeting applicable SLA for task turnaround (along with Regular Users maximized under DWA)**

RPA User Eligibility Decision Service

Generalized DMN Representation of Assignment Decision Logic (**)





RPAEligibility						
U	ComplexityRating	TaskTypeGroup	RPAUserSetting	RPAEligibility	Description	
	Input	*Input*	*Input*	*Outcome*	*Informational Note*	
1	"Low"	"ValidForRPAUser"	"RPAUserOnly", "RPAAndESWUser"	true	RPA online with qualifying task	
2	"High"			false	non-qualifying case	
3		"ValidForESWUser"		false	non-qualifying task type	
4			“ESWUserOnly"	false	RPA offline	

** Simplified logic done in DMN Model was created with the Trisotech DMN Modeler

RPA User Eligibility Decision Service

Managing Work Routing for Both RPA and Regular Users (*)

Appian User Group	RPA User Setting		
	ESW User Only	RPA User Only (Default)	RPA and ESW User
ESW User	Any eligible Tasks can be assigned.	Only eligible tasks marked for ESW Only (i.e., already reviewed by RPA User) can be assigned.	Any eligible Tasks can be assigned.
RPA User	No eligible Tasks can be assigned.	Only eligible Tasks not marked for ESW Only (i.e., not yet reviewed by RPA User) can be assigned.	Only eligible Tasks not marked for Regular User can be assigned.

RPA User Setting controls flow of work to the RPA User(s)

* DWA Admin Console was created in Appian

RPA User Eligibility Decision Service

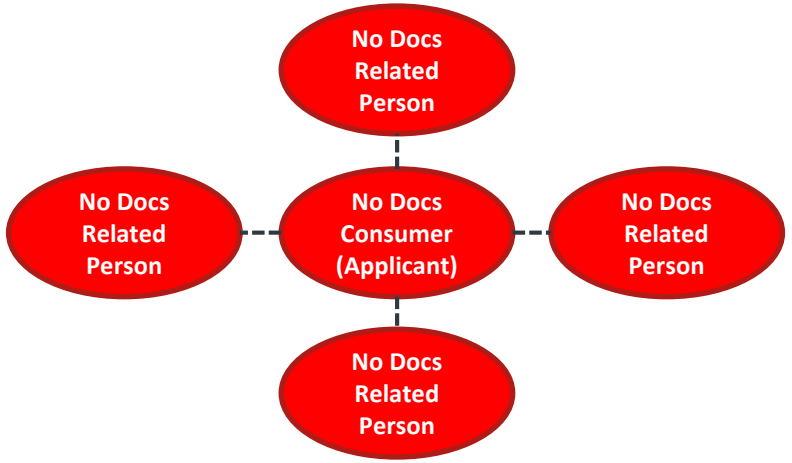
Low-Complexity Work Defined

Relevant Case File Data With Qualifying Issue(s) Pending Action

Qualifying Consumer's Application

+

Radio Silent Ever Status For Sending In Documents To Address Qualifying Issue(s)



=

Low-Complexity Work For Taking Pending Action (RPA User Work)

RPA User Service Design Approach

Treating RPA User Like Any Other User Resource

- **RPA User-eligible work is "low complexity" work, which the RPA User is able to work to completion after it interprets the evaluation of this possibility by the RPA Eligibility Service (as displayed for it to "see")**

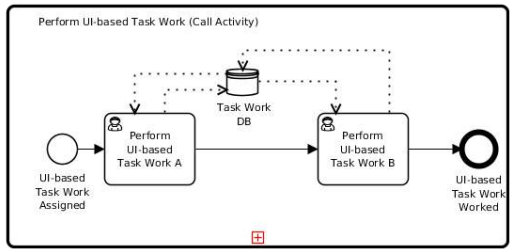
- **RPA User will do only what it is "taught" to do, which will be to follow the same set of actions/reactions to what is presented on screen as UI elements and behaviors that a Regular User would otherwise follow**

- **RPA User design path must also account for unusual contingencies, which are events it may encounter that a Regular User could otherwise work through (e.g., latencies in screen or UI element refresh)**

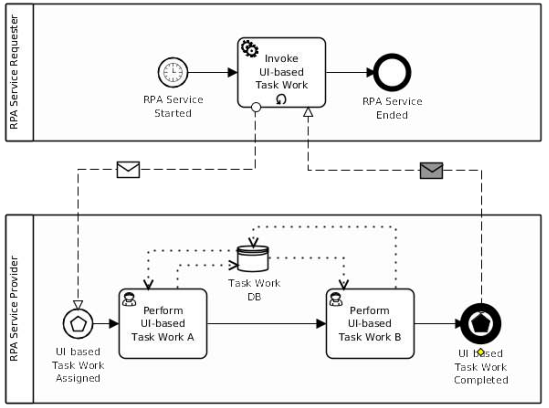
RPA User Service Design Approach

Modeling Limitations in Representing RPA User Service Launch

- RPA vs. Regular User launch the process in different contexts, which means the process is a Call Activity or an invoked via a Service Task…but are not all that helpful in modeling (though maybe in execution?)



Process as Call Activity that can be started within multiple contexts

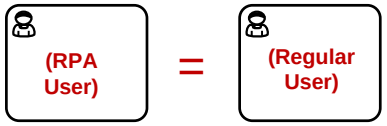


Process as a Service that can also be started by a Service Task

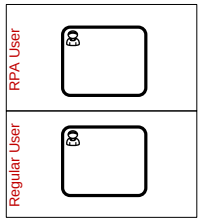
RPA User Service Design Approach

Modeling Limitations in Representing RPA User

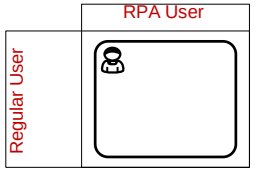
- **RPA vs. Regular User is really just a type of Performer or Role, which are attributes in BPMN and CMMN…but are not all that helpful in modeling (though maybe in execution?)**



Has 2 Performers but only 1 is in use at instance-level



Requires use of the User Task as a Call Activity



Semantics of this can be hard to explain to non-experienced modelers

RPA User Service Design Approach

Modeling RPA User Experiences as UX Design Patterns Instead

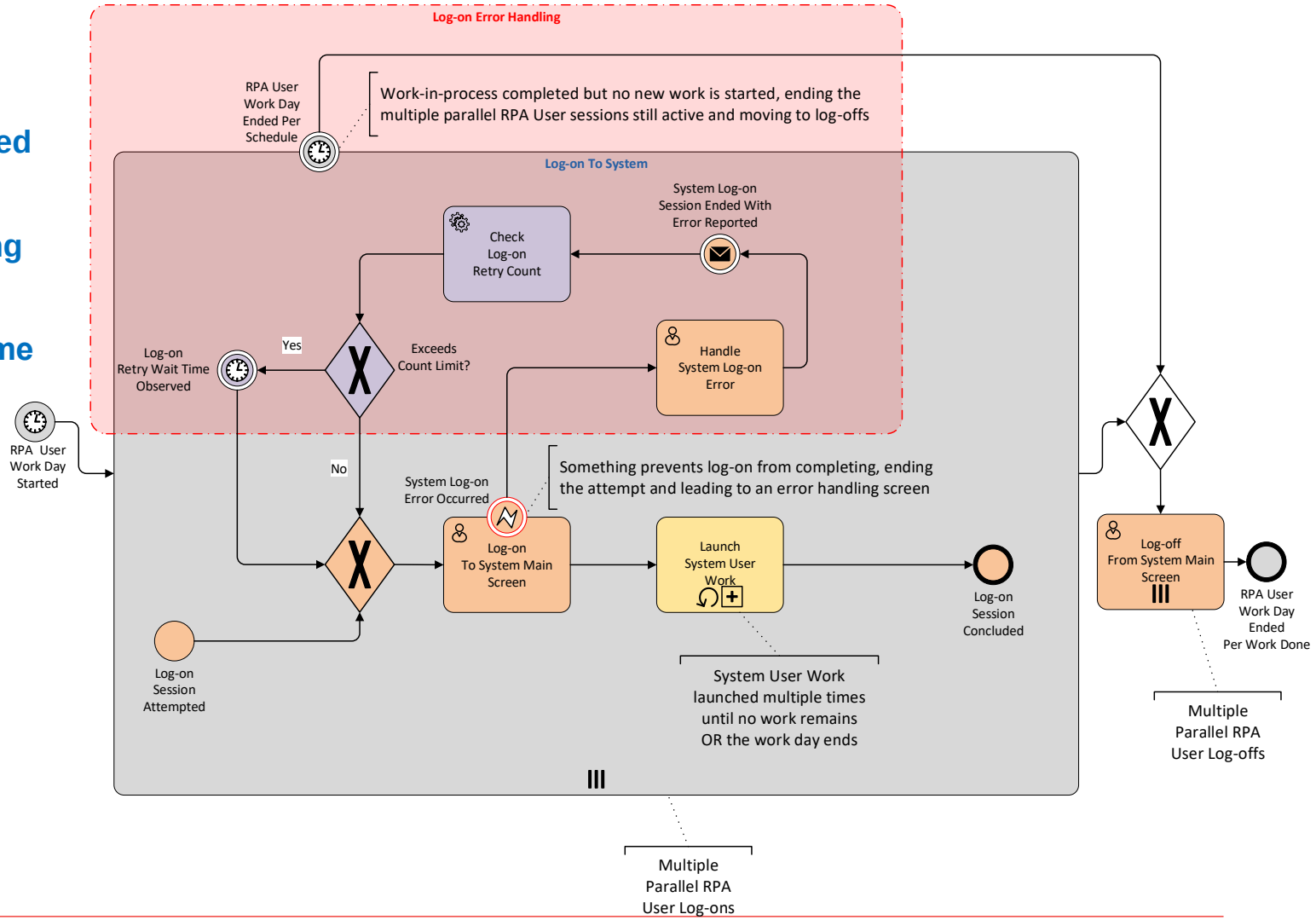
- ***Session UI initiation*** *– Logging on and handling any errors that are thrown*
- ***Work UI initiation*** *– Launching specific work screens and handling any errors that are thrown*
- ***Work UI navigation*** *– Navigating specific work screens along pre-defined lines of data entry or UI actions*

RPA User Service Design Pattern

Logging On To The Case Management System

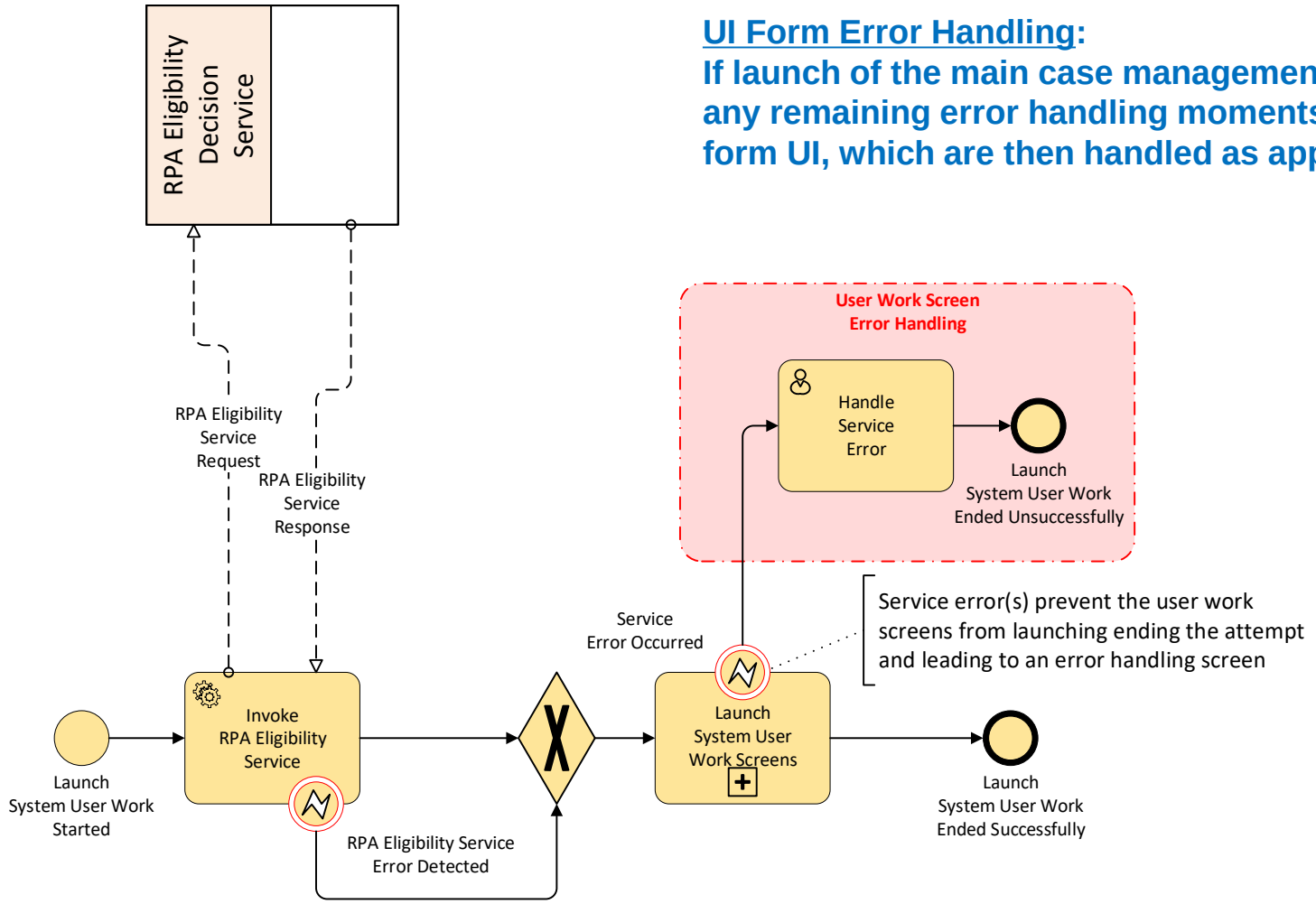
Logon Error Handling:

- **RPA User attempts to logon.**
- **If logon is successful, then Get Next Task is launched**
- **Otherwise (logon is not successful), and RPA User handles the error by closing the session and retrying if retry count is less than the pre-set limit.**
- **If the retry count is at the limit, then "sleep" for a time before retrying again with the retry count reset to 0.**



RPA User Service Design Pattern

Launching Process In The Case Management System

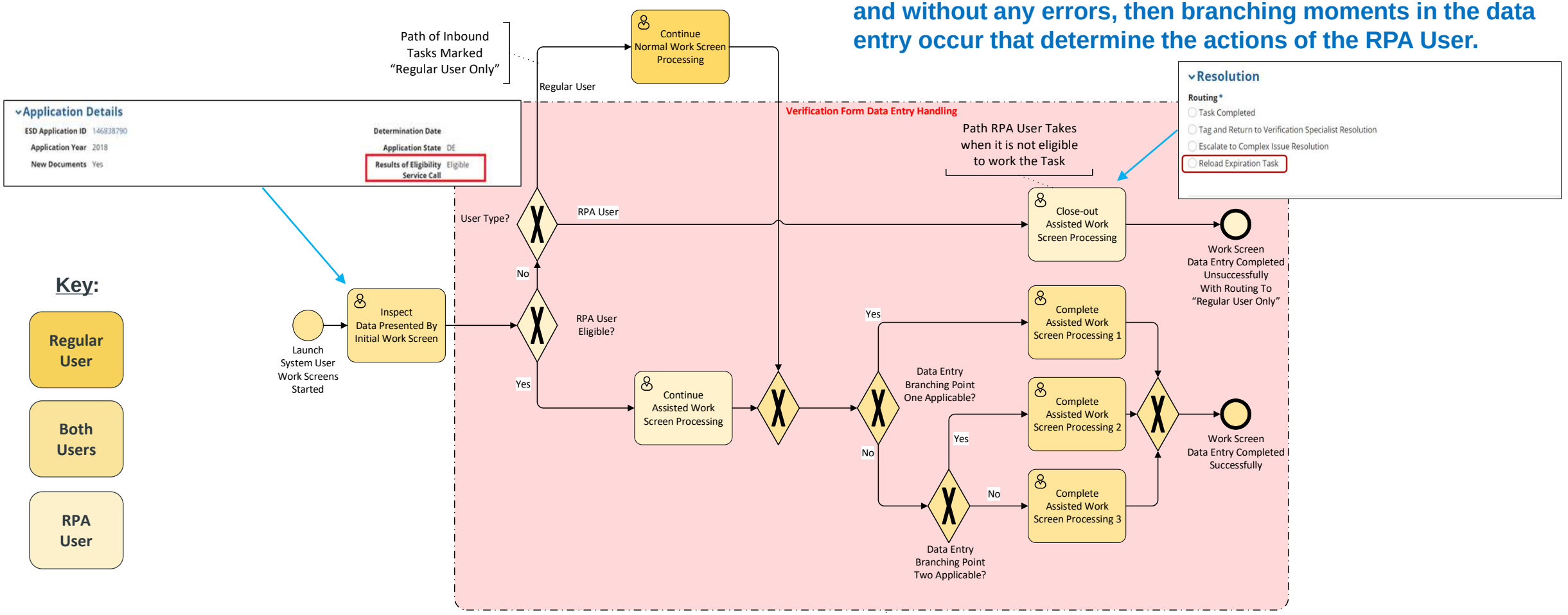


UI Form Error Handling:
If launch of the main case management process is successful, then any remaining error handling moments are presented on the returned form UI, which are then handled as appropriate by the RPA User.

RPA User Service Design Pattern

Entering Data Into The Case Management System

UI Form Data Entry Handling:
If launch of the main case management process is successful and without any errors, then branching moments in the data entry occur that determine the actions of the RPA User.



Lessons Learned and Looking Forward

Cultural Understanding/Acceptance of Using RPA Technology

- **Conditioned perception of what a "robot" does, which complicates its true understanding – e.g., errors the robot might make are seen as different though other users can make the same ones**

- **Lost in translation moments from SMEs to RPA designers, which occur because translating contextual decision-making informed through experience and training into algorithmic pathways can be hard**

Lessons Learned and Looking Forward

RPA Design Considerations

- RPA design works best with a fully deployable product, which means it such should not be done until after the system usage to be automated is more/less fully tested and stable

- RPA designer skills include understanding UIs from user perspective, which have much in common with automated functional testing tool scripters and those doing 508 Compliance work (assistive technologies)

Lessons Learned and Looking Forward

Future Applications of RPA

- 1st gen RPA designer tools use deterministic and structured flow modeling, which means the simple algorithmic pathways and low-complexity workstreams are first up

- Future gen RPA designer tools must include AI, which will enable more cognitively-dependent situations to be addressed by RPA Users

RPA-Enabled Case Management Solution

Demo Followed By Q&A

Demo Video Embedded Here