

## Technology Platform Team 1




# TPT.1



**NOTE:** This is a **technical** breakout session



# Agenda

- Geek Rating**
- Cast of **API Land** Characters..... .....
  - Elastic Search (**Dora & Neutron**)..... 
  - SAF Integration (**Perry**)..... 



**CWDS**  
Child Welfare Digital Services



# API Land Cast of Characters

Geek Rating



# API Land Cast of Characters



**Phineas**  
Intake UI Application



**Ferb**  
Intake API



**Dora**  
Search API



**Perry**  
Security API



**Barney**  
Legacy CRUD API



**Neutron**  
Search Engine Loading

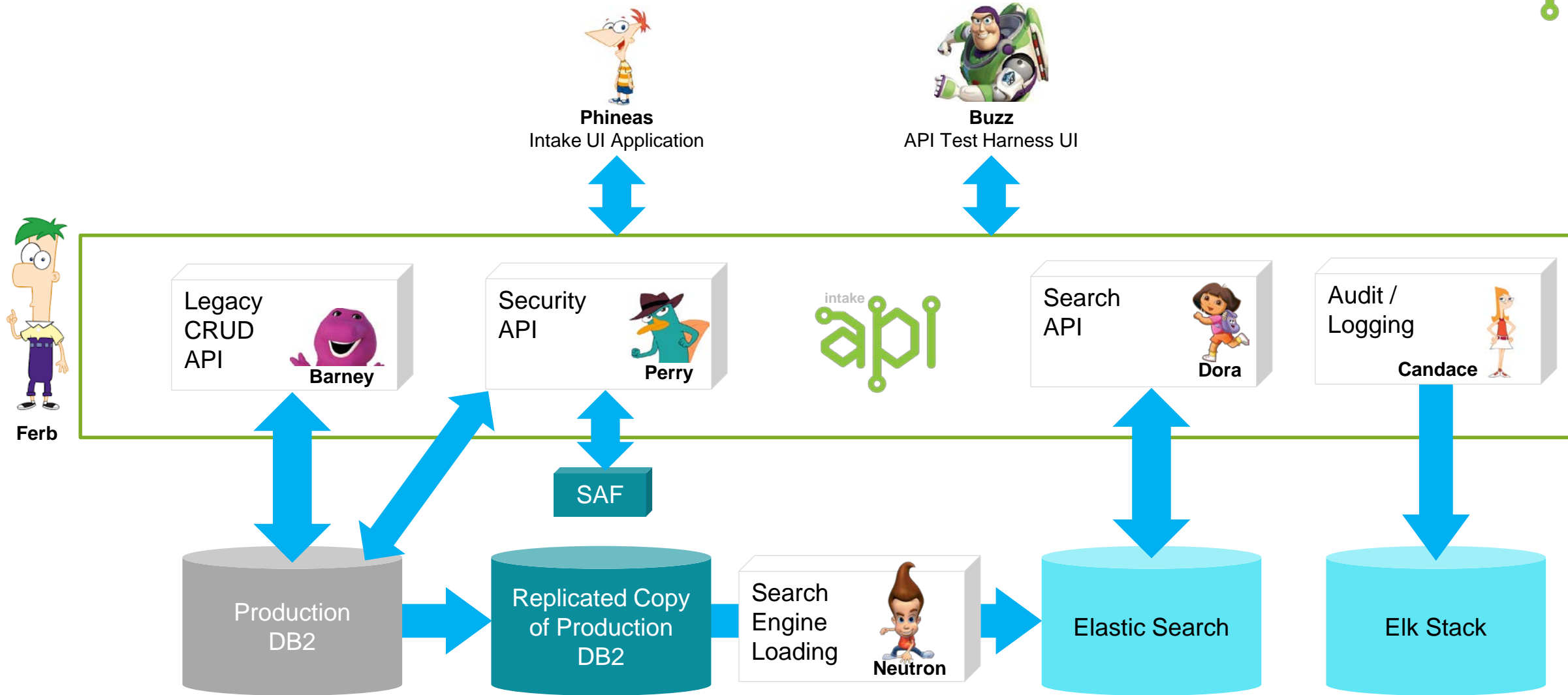


**Buzz**  
API Test Harness UI



**Candace**  
Audit / Logging

# API Land Cast of Characters





**CWDS**  
Child Welfare Digital Services



# Elastic Search

**Geek Rating**



**Neutron**

Search Engine Loading



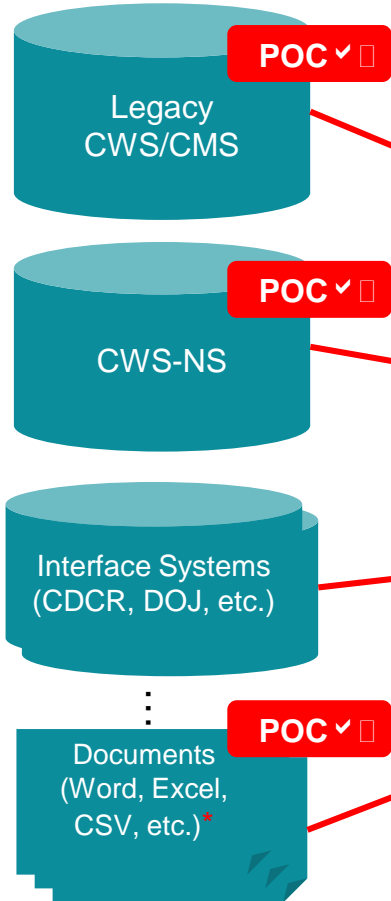
**Dora**

Search API

# Elastic Search: Overview

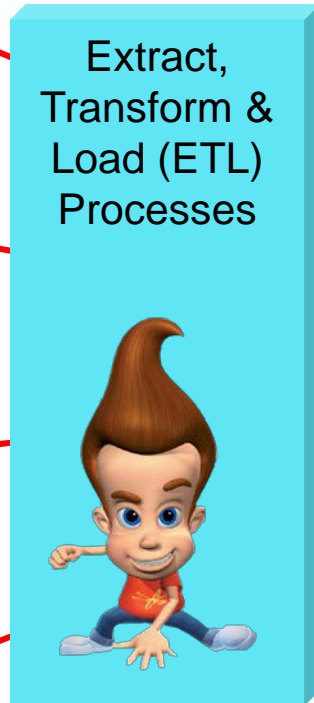
## Data Sources

Data can be pulled from a variety of disparate sources



## Data Loading

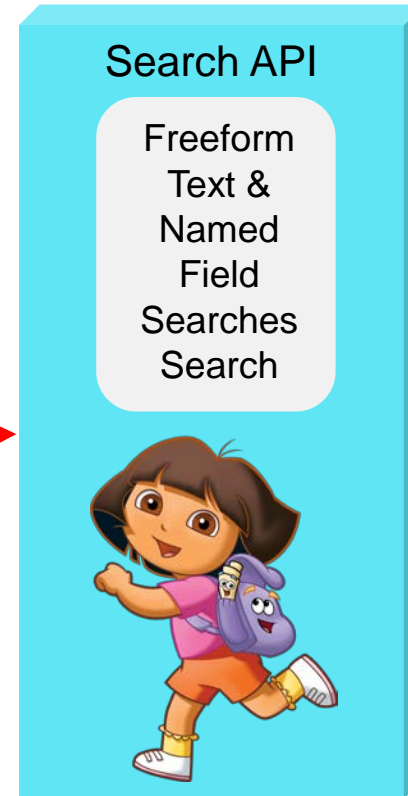
All data that needs to be searched against is loaded (copied) into Elastic Search, and continually updated.



Neutron

## Search API

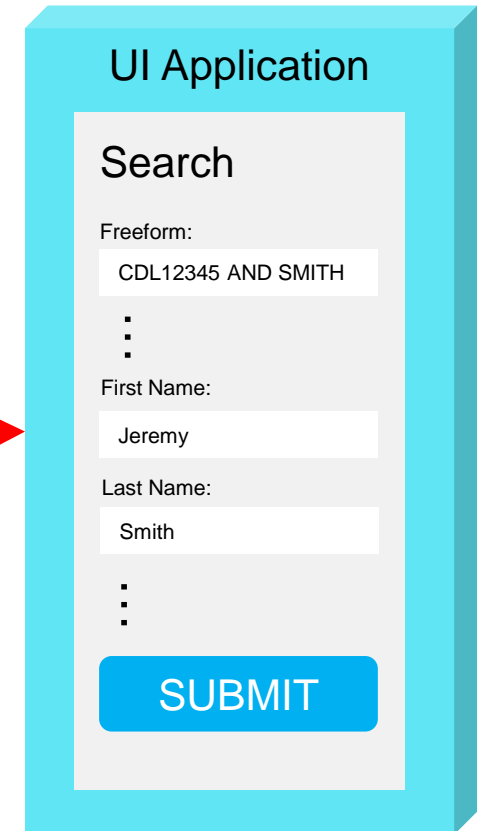
Search API provides search resources for searching freeform text (e.g. Comments, documents, etc.) and named fields (e.g. Last Name, First Name, Street Address, etc.)



Dora

## User Application

Application can present search UI as desired, and call Search API resource(s) to execute the search



# Elastic Search: Architecture / Performance

## Problem:

Initial loading of millions of records from Legacy Production DB2 into Elastic Search can be expensive from both performance and cost perspectives. Continual near-real-time updates would also be expensive.

## Pros:

- Single data source

## Cons:

- Potential performance degradation of Legacy Production system during initial loads and continual updates
- Adding table-level triggers and new tables (in order to determine which updates need to be loaded) would require lengthy and costly impact analysis and testing by the Legacy team





# Elastic Search: Architecture / Performance

## Solution (Implemented):

Use database replication to implement a read-only copy of Legacy Production DB2.

### Pros:

- Isolates ES loading and updates from actual Legacy Production DB2
- Provides ability to easily add table-level triggers and new tables (in order to determine which updates need to be loaded)

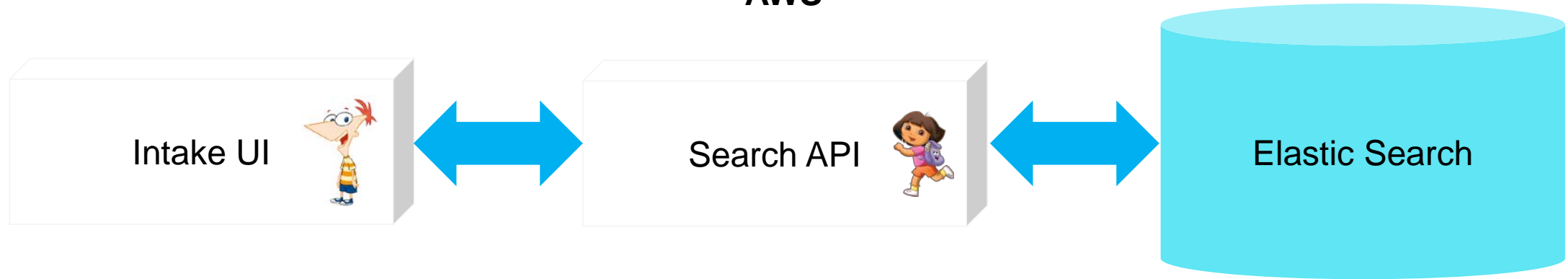
### Cons:

- Database replication needed to be installed / configured on mainframe

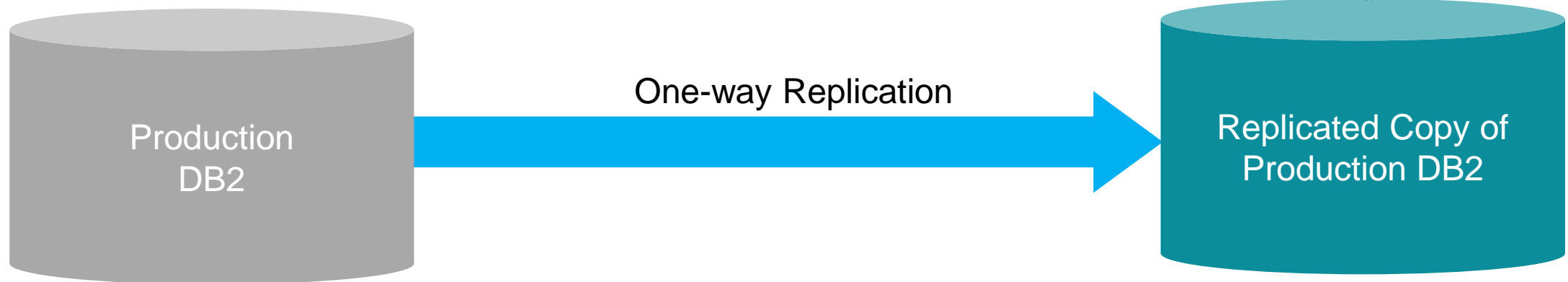


# Elastic Search: Architecture / Performance

## AWS



## Mainframe



# Elastic Search: Search Legacy Documents (attachments)

## Problem:

Legacy currently does not support searching within attached documents.

## Solution (POC implemented):

Mapper Attachments plugin for Elastic Search

<https://github.com/elastic/elasticsearch-mapper-attachments>

## Benefits:

- Allows searching against millions of document attachments from Legacy
- Supports many common document types: (PDF, Word, XLS, PPT, etc. – uses Apache Tika)
- Automatically determines common document metadata (Author, Document Name, Document Type, Keywords, Content Length, Last Update, etc.) and populates metadata fields in the Elastic Search index (also searchable)



# Elastic Search: Secure Pass-Through

## Problem:

Initial implementation of Dora was customized to take an input string, & return formatted results.

## Pros:

- Can easily lock down access to ES data via user authorizations
- Abstracts ES query syntax from UI application

## Cons:

- Requires code changes for any new query functionality (e.g. Highlighting)
- Hinders UI application developers from implementing features quickly



# Elastic Search: Secure Pass-Through

## Solution (in flight):

Implement a secure pass-through architecture, utilizing X-Pack security to lock down access.

### Pros:

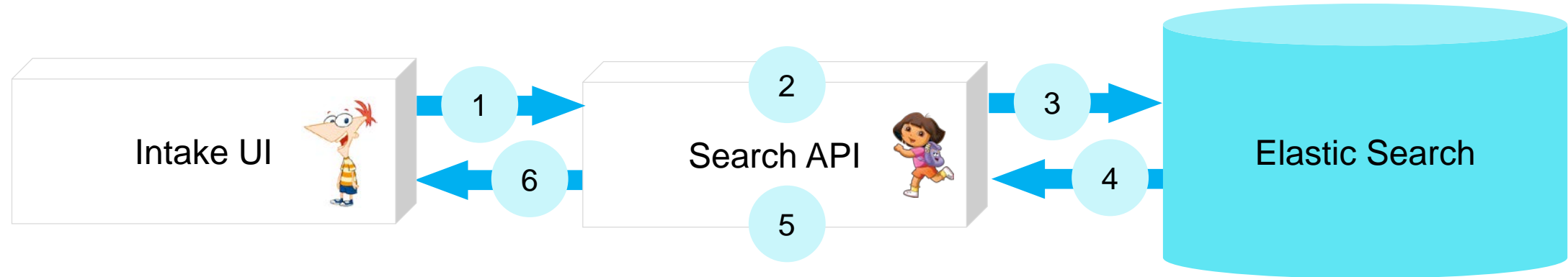
- Can lock down access to ES data via user authorizations
- Allows UI application developers to use regular ES query syntax (help, training, and examples are easy to find on Internet)
- UI application developers can implement new features quickly
- New features (like highlighting) will not necessarily require code changes to Dora

### Cons:

- ES Query syntax knowledge necessary for UI application developers



# Elastic Search: Secure Pass-Through



1. UI application passes Elastic Search (ES) query to Dora via json document using native ES syntax
2. Dora determines user authorizations
3. Dora submits query to ES
4. Dora receives results from ES
5. Dora filters results based on user authorizations
6. Dora returns results to UI application





**CWDS**  
Child Welfare Digital Services



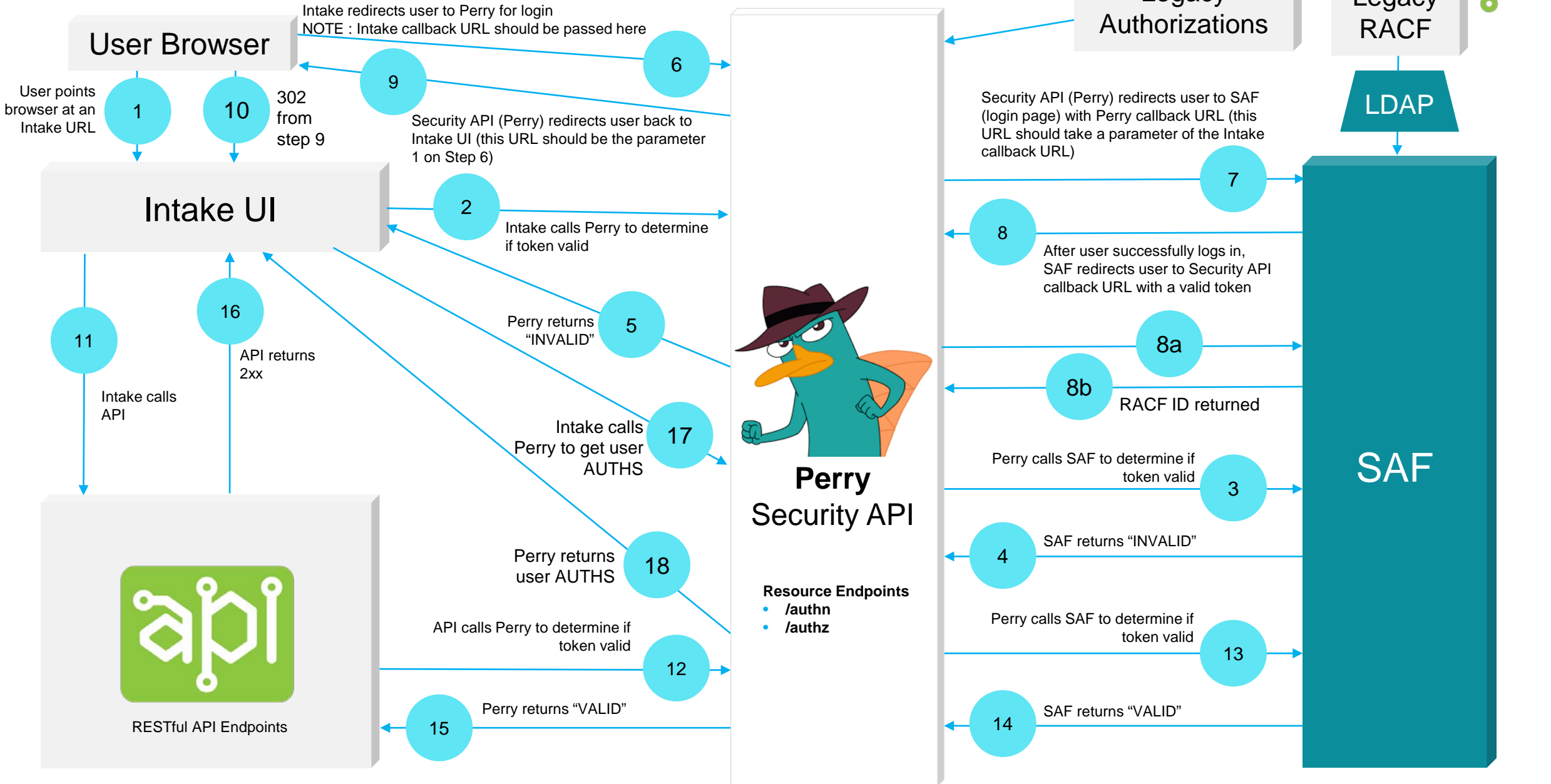
# SAF Integration

Geek Rating



**Perry**  
Security API

# SAF Integration





## Technology Platform Team 1

# TPT.1

Richard Bach[TPT.1 Technical Lead]

Gregg Hill[TPT.1 Scrum Master]

