

SUNNY SINGH'S PORTFOLIO

Currently EA lead at Amex for :

- ELF (Enterprise Logging Framework)
- EDV (Enterprise Data Vault)
- CAS ESI (Enhanced StandIn)
- EAG (Enterprise API Gateway)
- EWP (Enterprise Web Proxy)
- Risk Decision Tech Portfolio: CAS POA & RDM 2.0
- GCS Portfolio: CPG, oneAP
- Spring Boot Native, Graal VM & Knative
- Hybrid Cloud Data Strategy
- All EA Frameworks Support & SRE

Tech Expertise :

- Distributed Computing
- Clustered Messaging
- Data management
- Distributed Databases
- Low Latency, High Volume Computing
- Mechanical Sympathy
- Cloud Native Technologies
- CICD, Release Management
- Observability
- HA & Region Failover
- Linux, Java, C



REAL TIME OS (RTOS) & SEMAPHORES

R&D Automation EDN

RTOS (Real Time Operating System Software) & Semaphores

To reduce cost and use existing IoT devices in production, developed a RTOS that would replace Phillips PSOS (Phillips RTOS) to run control tasks to control machines and for real-time Observability & Alerting to take corrective action for machines in a power plant ([Realtime Observability Platform for machines](#))

In power plants and other industries, signals are collected by various IoT devices to control and tune the operation of a power plant and save machines by responding to them to shutdown in microseconds in real time. For example, when an equipment's temperature crosses a certain threshold, the equipment must be shutdown. This must be done in microseconds to save the equipment. (Realtime Observability Platform for machines in power plant)

Collaborated with a h/w engineer who designed a multi processor board with 8051 micro controller and Motorola 68000 processor. 8051 microcontroller would collect signals from the IoT devices and communicate with Motorola 6800 via DPRAM (Dual Port RAM). 68000 processor would process the data by running control programs to generate commands in real time to control and tune the operations of the devices that were being monitored. These commands were communicated using the Dual Port RAM which 8051 controller would read and send signals back to the device being monitored, to either control them or shutdown before they would be destroyed or become faulty.

This resulted in hardware & software cost reduction of DIPAS (Distributed Industrial Plant Automation System) by 10x

[I developed](#)

- [a RTOS \(Real Time Operating System\) that included a Realtime Task Scheduler and memory manager to manage control tasks/applications that run on Motorola 68000 ensuring deterministic response times in microseconds to control the devices being monitored](#)
- [Semaphores to protect simultaneous real-time Read and Write operations by 8051 microcontroller and Motorola 68000 processor to the Dual Port Ram](#)



INTERFERENCE & LICENSING SOFTWARE IRIDIUM CELLULAR SYSTEM

Motorola

Interference & Licensing software Iridium Cellular System

Developed and delivered Interference & licensing software for Iridium to honor frequency localization laws in certain jurisdictions around the globe. ([Radio Frequency localization laws](#))

Iridium cellular system is a constellation of 66 low orbit non geosynchronous stationary satellites. This allows Iridium phone to directly communicate with these low orbit satellites avoiding the need for cell phone towers and thereby allow cell phones to work anywhere in the world. The Iridium satellites route a call through the satellite constellation. The constellation operates in a near circular Earth orbit at an altitude of 778 km. This system is coordinated by 12 gateways distributed around the world. Iridium satellites uses an array of spot beams to communicate with subscriber handsets. The user dials a telephone number with the handset using an international 13-digit number as one would normally do using a standard phone. The user presses the “send” button to access the nearest satellite. The call is routed through the constellation and drops directly to the Iridium handset or to the closest gateway closest to the destination for non-Iridium handsets where the call is completed over standard circuits.

Radio frequency bandwidth is a scarce commodity. Communication systems like Iridium create interferences to other radio equipment and telecom operators in certain jurisdiction. There is a need to beam signals using a frequency that does not interfere with other systems and/or NOT beam the signal at all due to geopolitical conflicts (like in Iraq, Iran, North Korea, ...))

[I led a team of approximately 30 developers in US and India to develop](#)

- [Interference and licensing software that ran on the satellites and gateways to not beam signals or beam signals using a licensed frequency to honor the localization laws in certain jurisdictions around the world. \(Radio Frequency localization laws\)](#)



CT2 PROTOCOL & CDMA CELLULAR CALL HANDOVER

Motorola

CT2 Protocol & CDMA cellular call handover software

Developed and delivered CT2 protocol and CDMA/GSM cellular call handover software

CT2 involves replacing telephone booths with wireless base stations. Customers buy inexpensive CT2 hand help phones and could make calls anywhere within base-station range.

In CDMA, cell phones talk to the cell tower which in turn talk to the communication satellites. This signal is then beamed to other towers which in turn connect to the cellphones where the call terminates.

When the customer travels while talking using CT2 handset or a CDMA cellphone, a running call must seamlessly mover over to connect to another base station or tower to avoid any disruption of service for the ongoing call. This is called call handoff/handover.

I developed

- the CT2 protocol between base station and CT2 phones
- Call handover software in cell phones which hands off a running call from one cell tower / base station to another when a customer is moving in a car (say). This avoids any disruption of service for an ongoing call



TUXEDO & QUEUEING ENGINE DEC MESSAGE Q

BEA Systems (Acquired by Oracle)

Tuxedo (Transactions For Unix, Extended for Distributed Operations) and DEC Message Queue

Developed and delivered Tuxedo /T , /Q and DEC Message Queue

Tuxedo (Transactions for Unix, extended for Distributed Operations) is an application server to provide extremely scalable applications and High Availability to support applications requiring thousands of transactions per second on commonly available hardware. Tuxedo provides /T and /Q subsystems. Tuxedo /T is TP Monitor that coordinates XA transactions spanning multiple XA resources. /T also provides platform for deploying application functions as a service. Tuxedo /Q is a queuing engine to support messaging. Tuxedo till date has a revenue of ~2B dollars.

DEC Message Queue is messaging middleware acquired by BEA Systems from Digital

I developed

- Enhanced Tuxedo Store boot times from couple of hours to seconds
- Tuxedo library, file XA Resource, platform for function as a service, Queuing Engine
- Queuing Engine rewrite for DEC Message Queue (which was rebranded as AMQ – Advanced Message Q)



WEBLOGIC SERVER

BEA Systems (Acquired by Oracle)

Implemented or lead implementation of

CLUSTERED DISTRIBUTED MESSAGING

JMS

WRITE AHEAD LOGGING STORE

NOSQL STORE

RE-ENTRANT FINITE STATE MACHINE

JMS XA RESOURCE

PERFORMANT GUIDS

CUSTOM JAVA SERIALIZATION

OBJECT VERSIONING

UNIT OF ORDER

STORE AND FROWARD

SPLIT BRAIN

BOXCAR WRITES

READ REPLICAS PERFORMANT COUNTERS

EXALOGIC

SEARCH

TELEMETRY

Weblogic Server

Author of multiple components in Weblogic Server over a span of 13 years

Weblogic server is a platform for developing, deploying and running enterprise applications, such as Java, for on-premises and in the cloud. Weblogic server offers implementation of Java Enterprise Edition. Weblogic server is fully supported on Kubernetes and enables users to migrate and build modern container apps with Java service. Oracle purchased Weblogic server. Weblogic still runs today.

Oracle has a patent for store and forward messaging. I am listed as one of the 2 inventors in this patent

I developed and led teams to deliver Weblogic Server over a span of 13 years

- Implemented Clustered Distributed Messaging Core to support internal communication of services and foundation for Java Messaging Service in Weblogic Server
- Implemented Java Messaging Service (JMS) in Weblogic Server
- Implemented Write Ahead Logging Store for Weblogic Transaction XA Coordinator and JMS store. As part of this work, discovered that writes suffered on an average wait times of one half rotation of disk for any given write. Implemented techniques to find rotational latency of disks in Java. This helped to align the disk heads and disk location when writing to disk and thereby avoiding disk slipping and write operations to wait for one full rotation
- Was part of team that conceptualized and implemented CI/CD tools – Build, Test & Deploy Monkeys inside BEA systems. Members of the team went on to deliver CruiseControl and Jenkins in the Open-source community

Weblogic Server (Continued)

Author of multiple components in Weblogic Server over a span of 13 years

- Implemented no SQL Store for Weblogic Server
- Implemented re-entrant Finite State Machine pattern for Weblogic server to be used by various subcomponents internal to Weblogic Server. This was the core that helped Weblogic server to be scalable and performant when compared to its competitors. Parts of this work were later seen to be incorporated in JDK Futures and Reactive APIs
- Implemented GUIDS that are statistically unique in space and time and 5X faster when compared to OOTB GUIDS provided by the JDK. These GUIDS were used as message IDs for the messaging and transaction subsystems in Weblogic Server
- Implemented Custom serialization of Java Objects, as Java Serialization provided by Java in the JDK is slow
- Implemented JMS file XA resource that can be infected with XA transactions started by the XA Coordinator
- Implemented Object versioning framework to make every internal Java object in Weblogic server to be backward compatible

Weblogic Server (Continued)

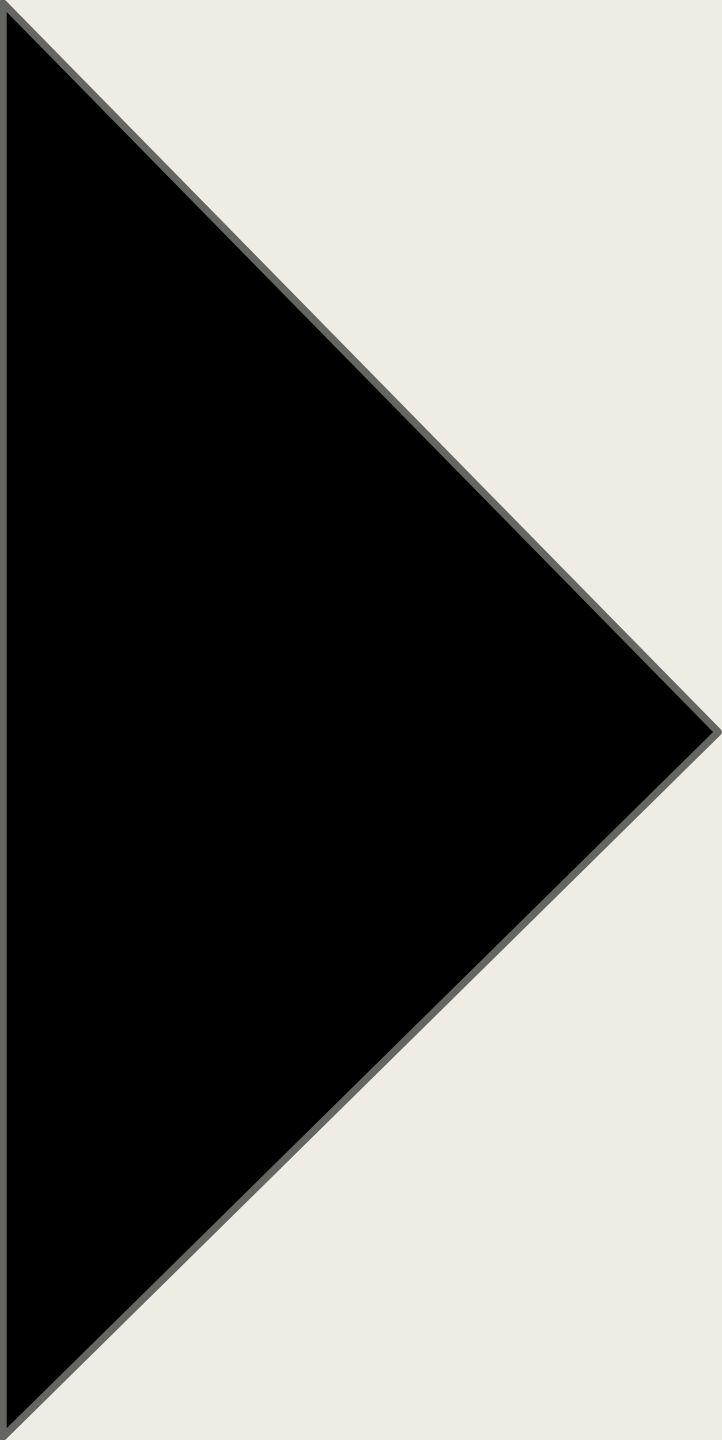
Author of multiple components in Weblogic Server over a span of 13 years

- Implemented Unit Of Order for Distributed Clustered Topics. This allows a given group of messages to be processed by one given consumer in the Weblogic cluster. This feature facilitates multiple consumers on a clustered distributed topic and still maintains data integrity.
- Implemented techniques to detect Split Brain in Distributed Queues and Topics in Weblogic Cluster
- Implemented Boxcar Writes for File and Object Store to make them more performant
- Implemented read replica striped Java counters to avoid performance bottlenecks caused by statistics subsystem
- Implement Store and Forward service for Weblogic cluster
- Implement multi tenant messaging service in Cloud
- Managed Search and Telemetry services for Oracle Cloud

Weblogic Server (Continued)

Author of multiple components in Weblogic Server over a span of 13 years

- Implemented techniques to increase messaging throughput from 25K messages/second to 125K messages per second in Weblogic server on Exalogic running Linux. Identified network interrupts were not load balanced across cores in a multicore machine and were pinned to one core. Worked with the Linux kernel team to implement software interrupt load balancing to multiple cores in the Linux kernel. This has been incorporated into Linux kernel since then.



QUANTUM TRADING PLATFORM

Citigroup

Quantum Trading Platform

Global Head Quantum Platform used by Equities and Fixed Income in Citi

Was hired to remove inefficiencies in high-speed trading for Equities at Citi. After spending time understanding the problem domain for about a year, bootstrapped the development and operationalization of Quantum Trading Platform that was Asset neutral and used by Algo, Program, Sales and High Frequency Trading. I managed directly and indirectly around 1000 colleagues in US, UK, India and Japan

Thought Leader for Citigroup Quantum Trading Low Latency Trading Platform with peak capacity of 120K messages/second and daily transaction volume of 2 Billion dollars for DMA, Algo, Cash and Market Making

Technology Evangelist responsible for establishing CICD and Agile Development Practices at Citigroup

Built highly performant team from within and outside Citi to build this product. This product delivered

- Mechanical Sympathy highly performant platform. This platform uses data structure alignments to allow threads to update variables and avoid CPU cache lines false sharing. False sharing occurs when threads on different processors modify variables that reside on the same cache line. This invalidates the cache line and forces an update, which hurts performance. This platform allowed for horizontal scaling, lock free algorithms, network packet & CPU core load balancing, blocking and non-blocking IO (futures & async future), Consistent Hashing for Trading Platform using distributed computing technique
- Reduced NUMA cache coherency issues across cores in CPU banks by pinning all threads in a thread pool to a set of cores that do not cross CPU bank boundaries (in other words in the same CPU Bank)
- Rules Engine for managing, splitting and filling orders
- Manage Grandparent, Parent and child orders by splitting orders as per the rules configured and models developed by Quants while managing the risk appetite and exposure to the market.

Quantum Trading Platform

Global head of Quantum Trading Platform used by Equities and Fixed Income in Citi

- Match executed fills from exchange and internal crossing engine to customer orders
- Fast Order Cache and Store
- Design and implement intelligent OrderIds that are statistically unique in space and time which were instrumental in operational observability of the platform
- Fast Buy Sell Order Book
- Crossing Engine to find dark pool and cross Buy And Sell limit orders internally. This avoids platform to send orders to the exchange and thereby saving exchange fees to cross orders
- Fill customer market orders at best price, as mandated by Reg NMS regulations
- Publish internal crossing price of a trade to the exchanges
- Manage low latency Market Data feeds from exchanges for High Frequency Trading
- End of day Reporting framework to balance the books
- Quantum FIX Engine connectivity to exchanges



ilogin.com

Founder

ilogin Direct 2 form factor login App

ilogin Secure Wallet & PayCard : https://www.youtube.com/watch?v=IG_ahORxWw8

ilogin Consumer Coupon App : <https://www.youtube.com/watch?v=S-07Q-GCsFk>

ilogin OTI Ticketing App : <https://www.youtube.com/watch?v=E7XGYJDnUn8>



Founder iLogin.com

Direct 2 form factor login

Developed direct 2 form factor login to reduce friction in 2 form factor login. iLogin powered web sites provide a QRCode for login (no username and password to enter on website). Users would scan this QR Code using their cell phone and enter their username and password in iLogin App. Upon successful login via cell phone, it would automatically sign in into the website on a computer. The intent is to avoid phishing attacks and use of public computers to login.

This did not become popular, as there was friction when scanning the QRCode



Founder iLogin.com

iLogin Secure Wallet & PayCard:

Developed iLogin secure wallet to load money into OTI NFC dongle. This dongle has an NFC element that can be carried in your keychain and used as a cash card. This dongle can be inserted into the ear jack of a cell phone (Android and iPhone). iLogin Wallet communicates to this dongle using audio frequency to load money into the dongle.

After the money is loaded into the dongle, one can unplug the dongle and carry it in their key chain and use it at any NFC enabled POS checkout counters. This wallet has been white labeled to OTI America and is currently used in Japan.

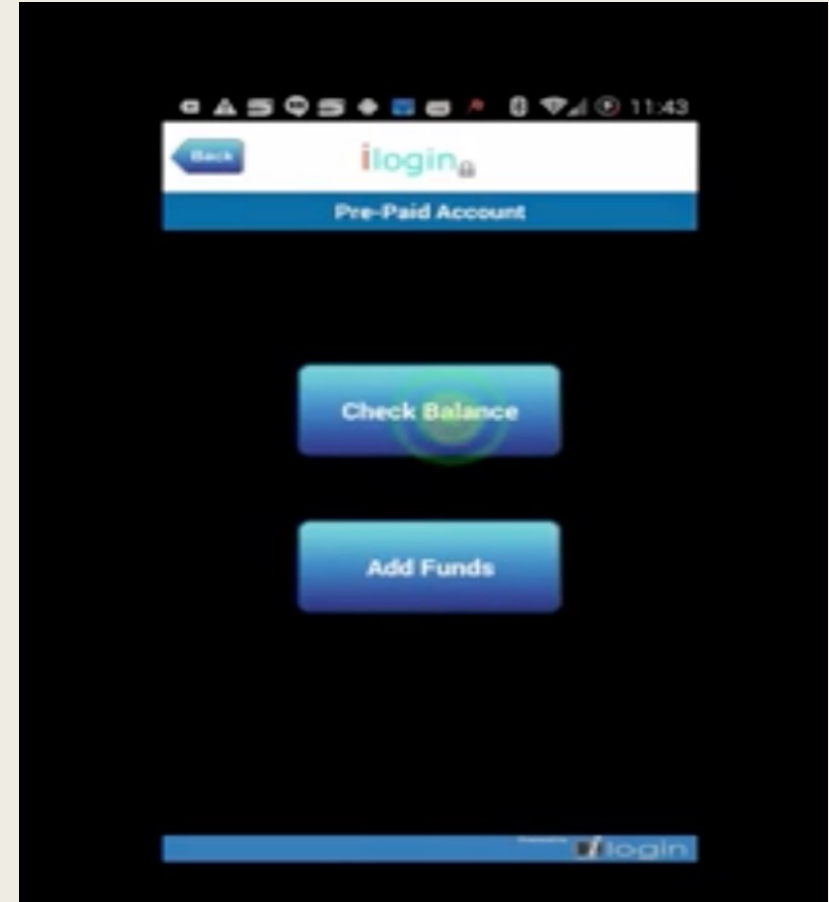
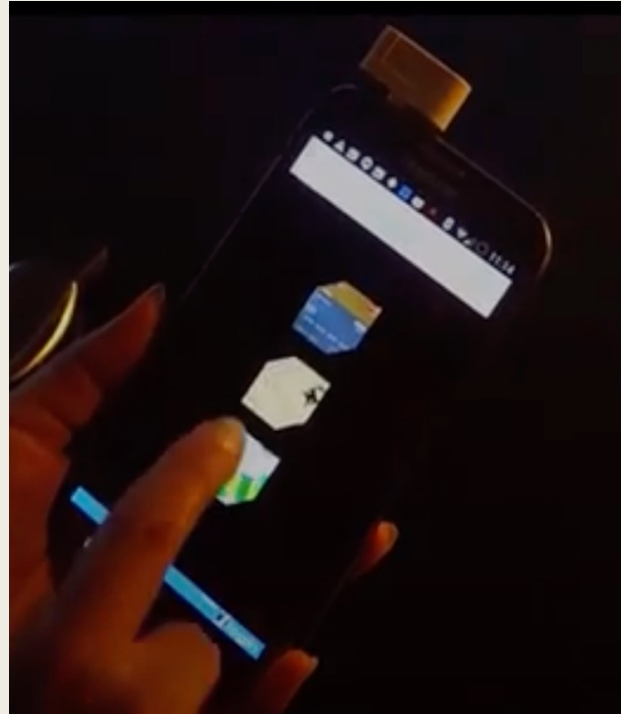
This uses the Split and Merge algorithm (patent pending) to secure login credentials and credit card numbers of the customer. The patent pending is also submitted to Amex when I joined.

https://www.youtube.com/watch?v=IG_ahORxWw8



Founder iLogin.com

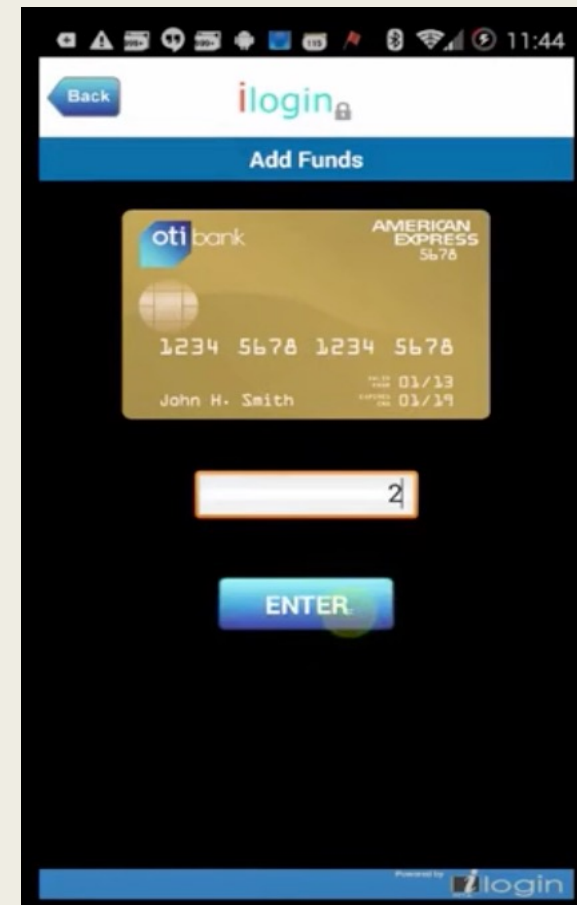
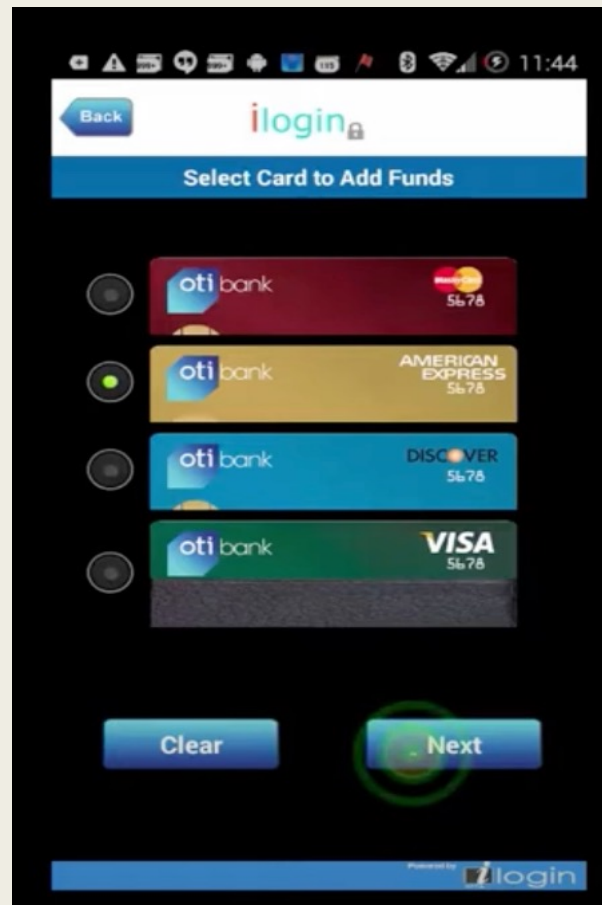
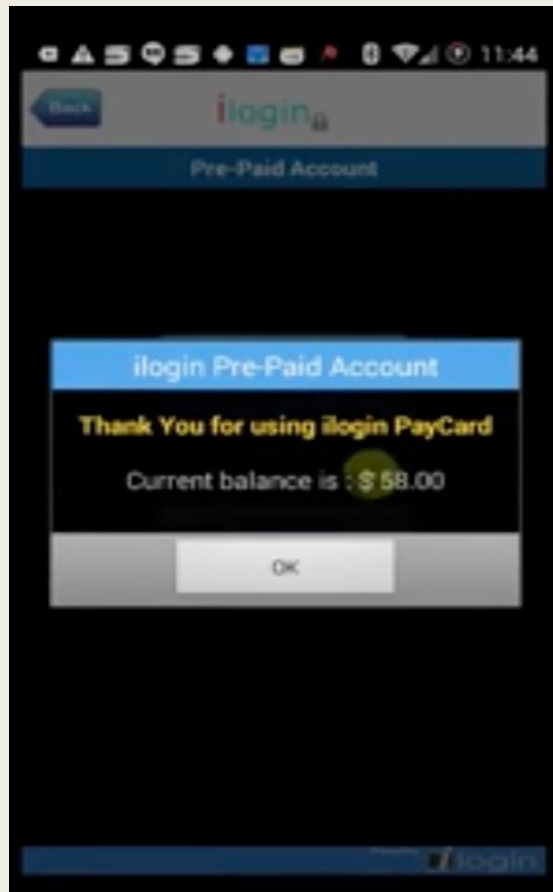
iLogin Secure Wallet & PayCard: https://www.youtube.com/watch?v=IG_ahORxWw8





Founder iLogin.com

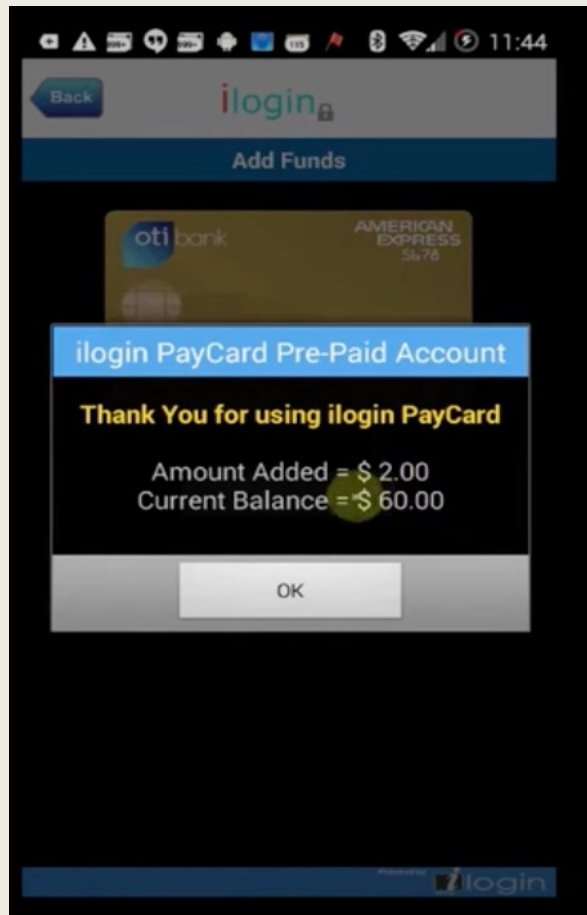
iLogin Secure Wallet & PayCard: https://www.youtube.com/watch?v=IG_ahORxWw8





Founder iLogin.com

iLogin Secure Wallet & PayCard: https://www.youtube.com/watch?v=IG_ahORxWw8





Founder iLogin.com

iLogin Consumer Coupon App:

Developed iLogin consumer coupon App. When a consumer sees a print media or an advertisement with a QRCode, iLogin app can be used to scan this QRCode. iLogin in realtime then communicates using merchant APIs. Merchants push items and coupons to the consumer iLogin App. Consumer can use this coupon at the POS checkout counter.

<https://www.youtube.com/watch?v=S-07Q-GCsFk>



Founder iLogin.com

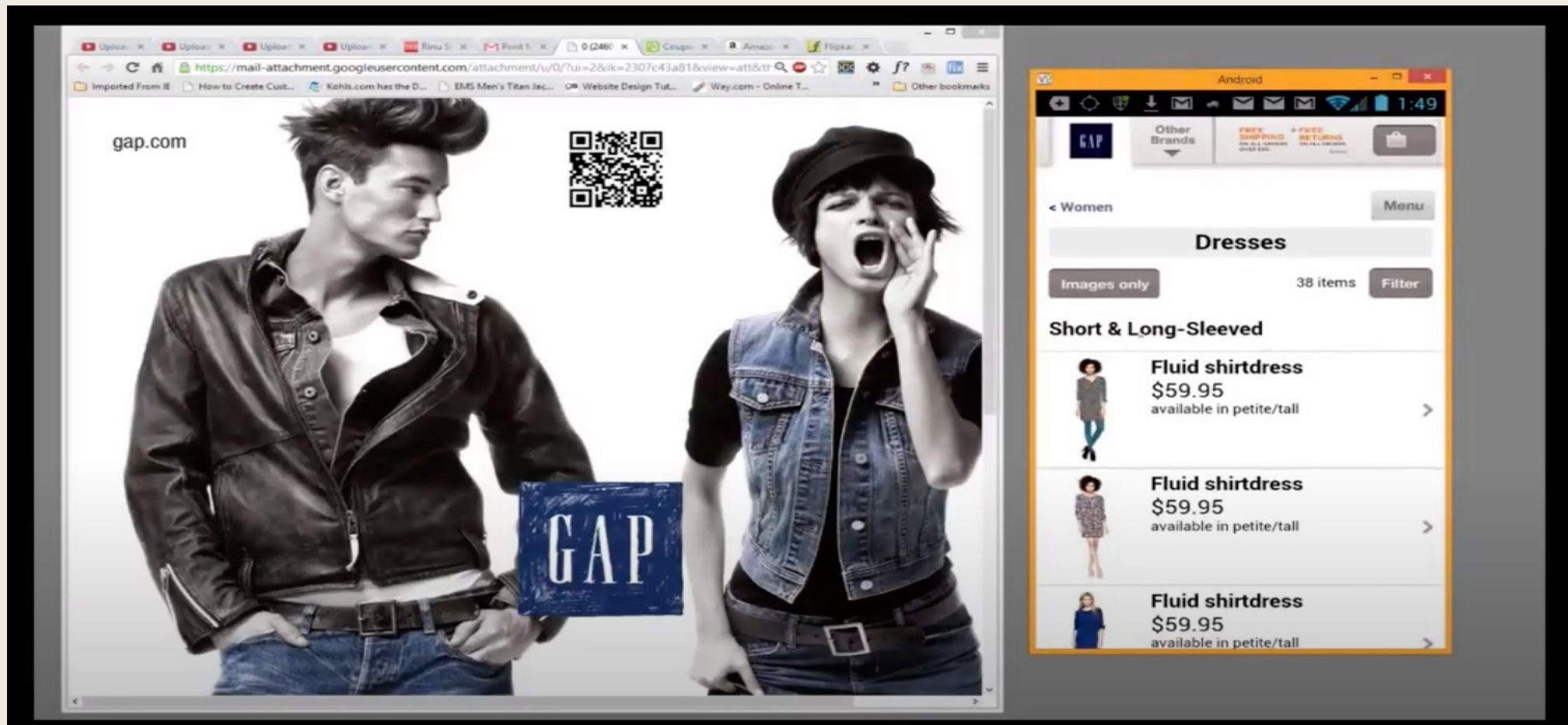
iLogin Consumer Coupon App: <https://www.youtube.com/watch?v=S-07Q-GCsFk>





Founder iLogin.com

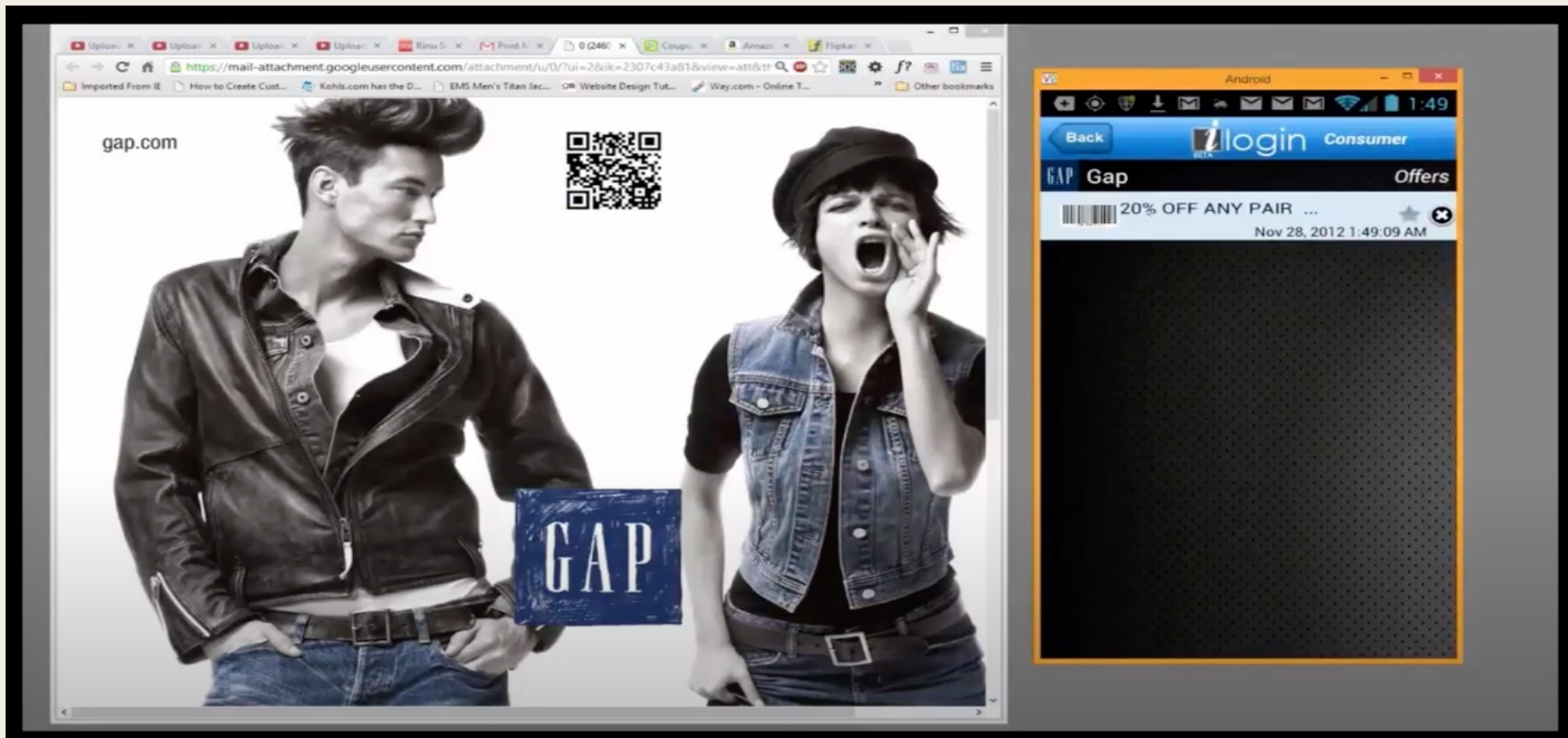
iLogin Consumer Coupon App: <https://www.youtube.com/watch?v=S-07Q-GCsFk>





Founder iLogin.com

iLogin Consumer Coupon App: <https://www.youtube.com/watch?v=S-07Q-GCsFk>





Founder iLogin.com

iLogin OTI Ticketing App:

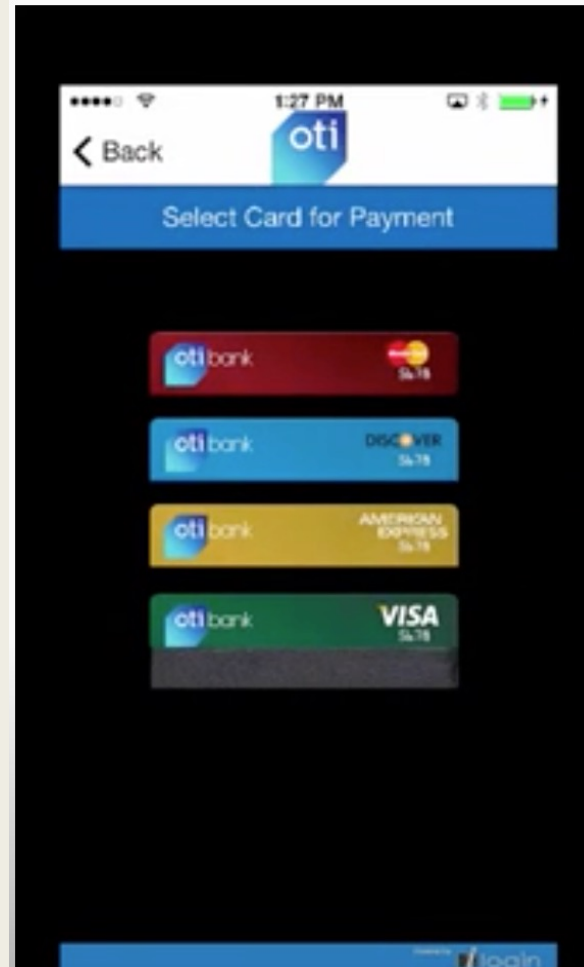
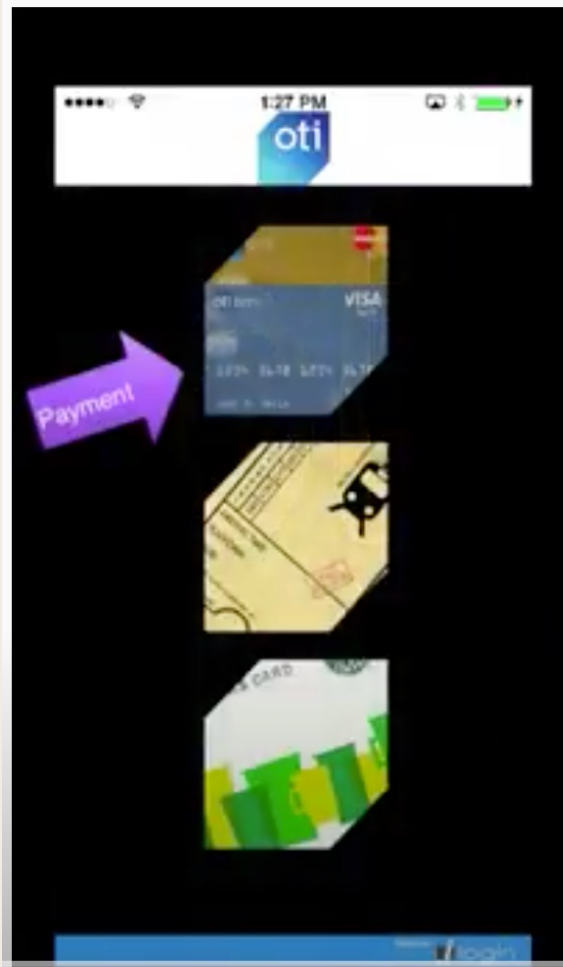
Developed iLogin NFC Wallet Ticketing App for NFC Phones. This App has been white labeled to OTI America. Users can load money into the wallet and use their phone to pay for tickets by tapping their NFC phone at the gates of Metro in certain cities of Europe

<https://www.youtube.com/watch?v=E7XGYJDnUn8>



Founder iLogin.com

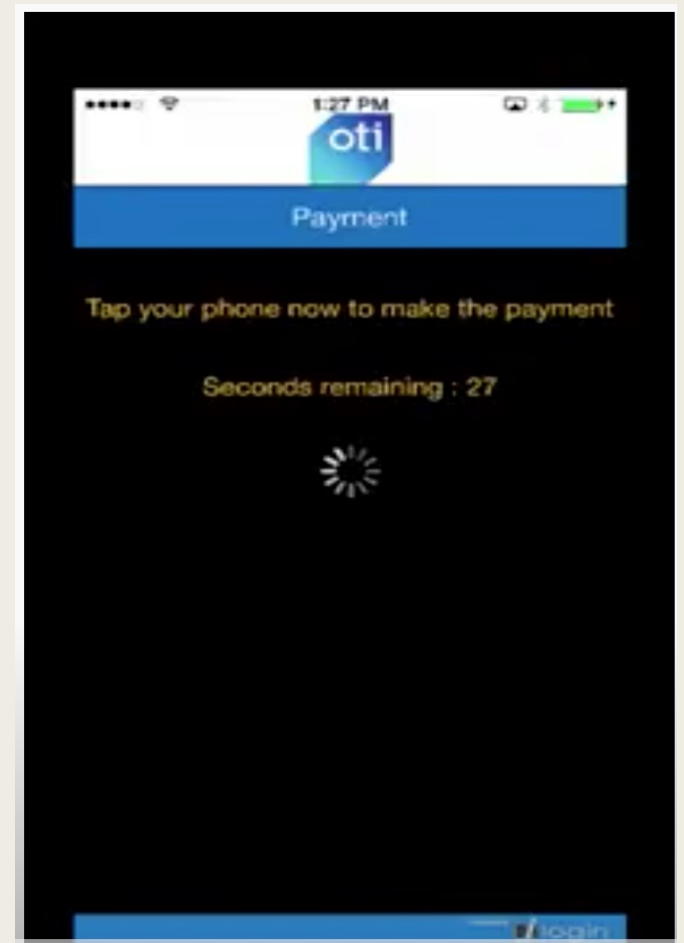
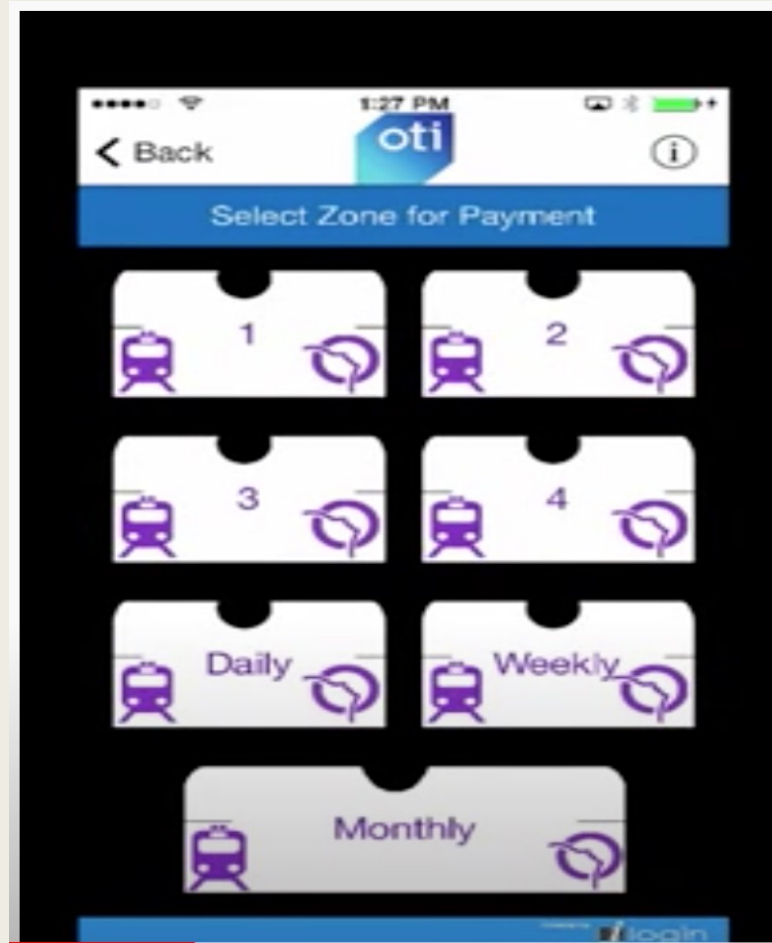
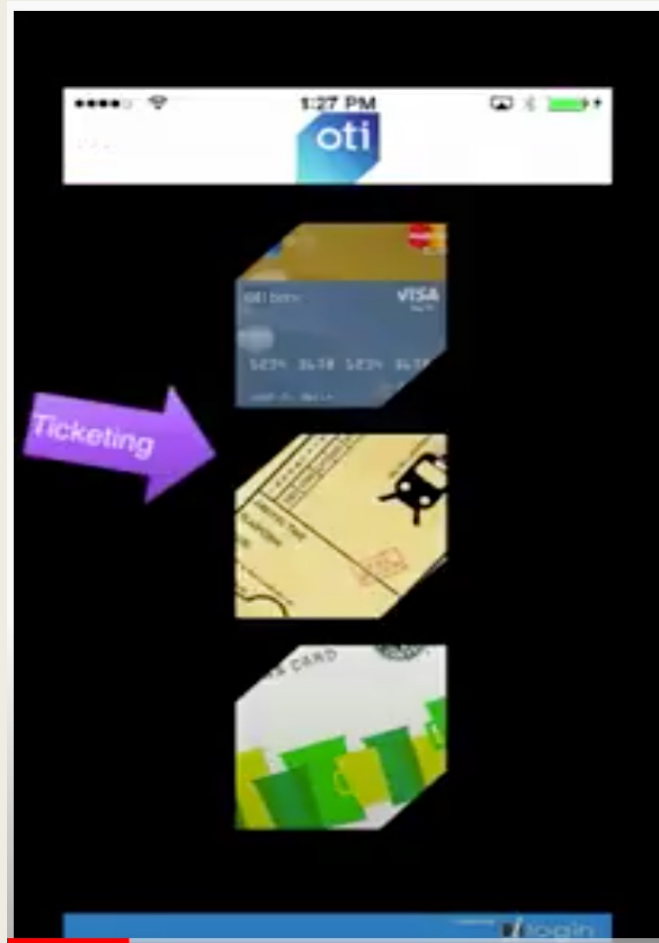
iLogin OTI App for NFC Phones: <https://www.youtube.com/watch?v=E7XGYJDnUn8>





Founder iLogin.com

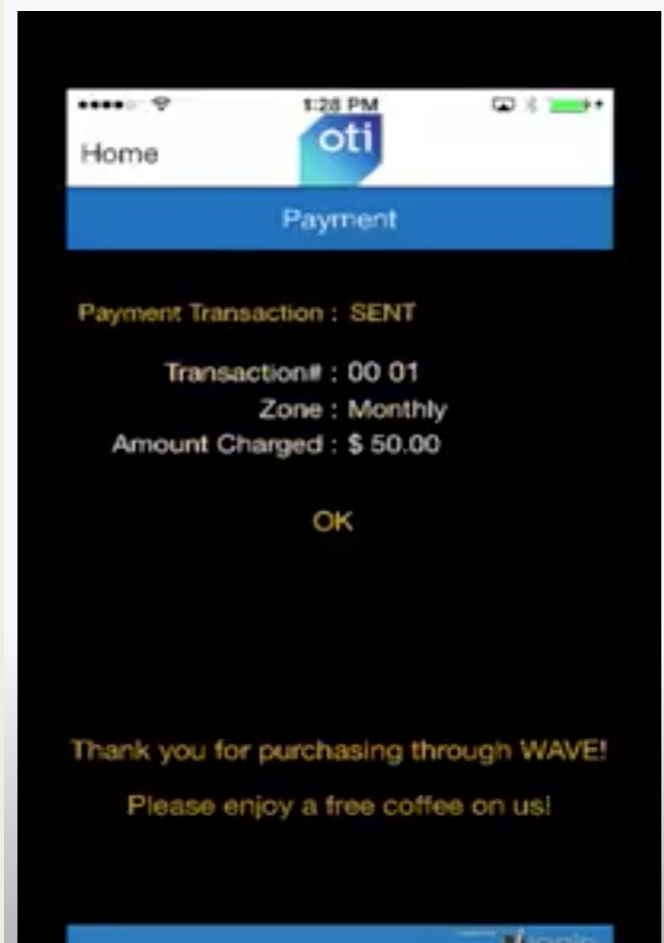
iLogin OTI App for NFC Phones: <https://www.youtube.com/watch?v=E7XGYJDnUn8>





Founder iLogin.com

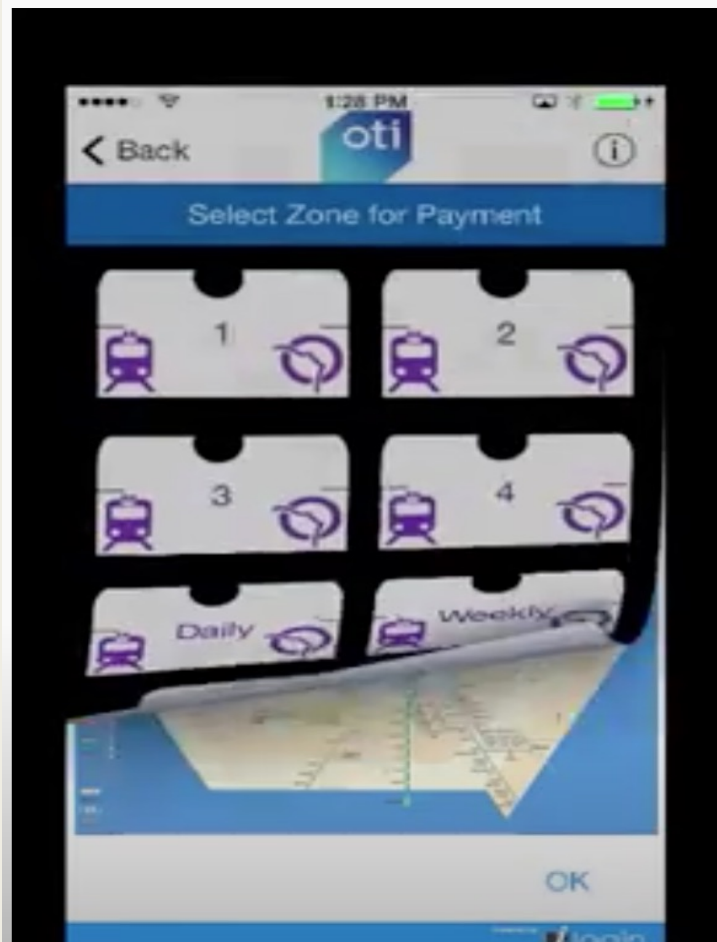
iLogin OTI App for NFC Phones: <https://www.youtube.com/watch?v=E7XGYJDnUn8>





Founder iLogin.com

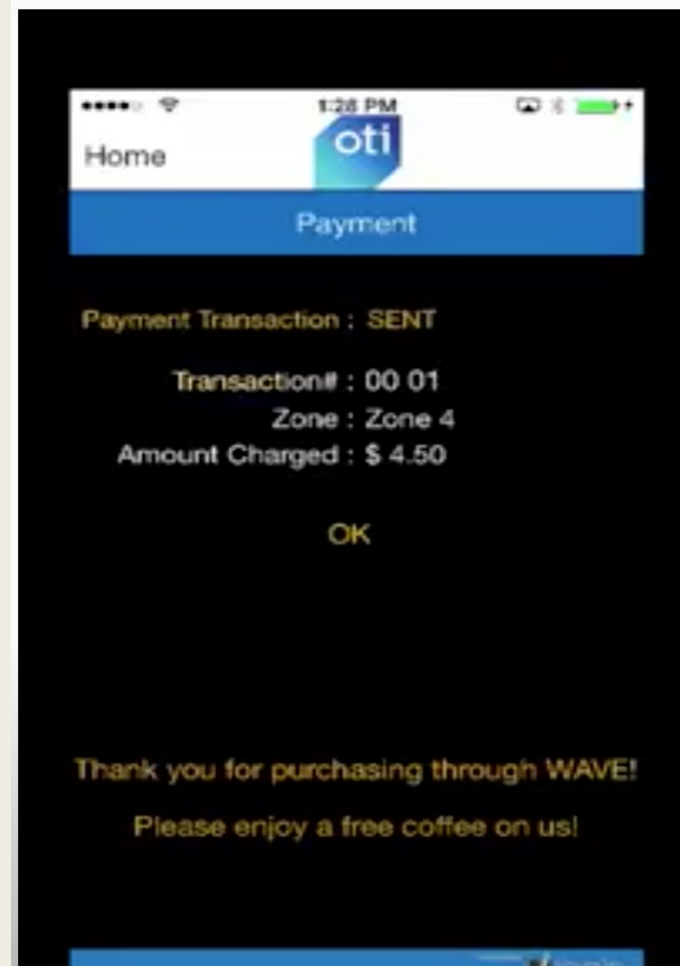
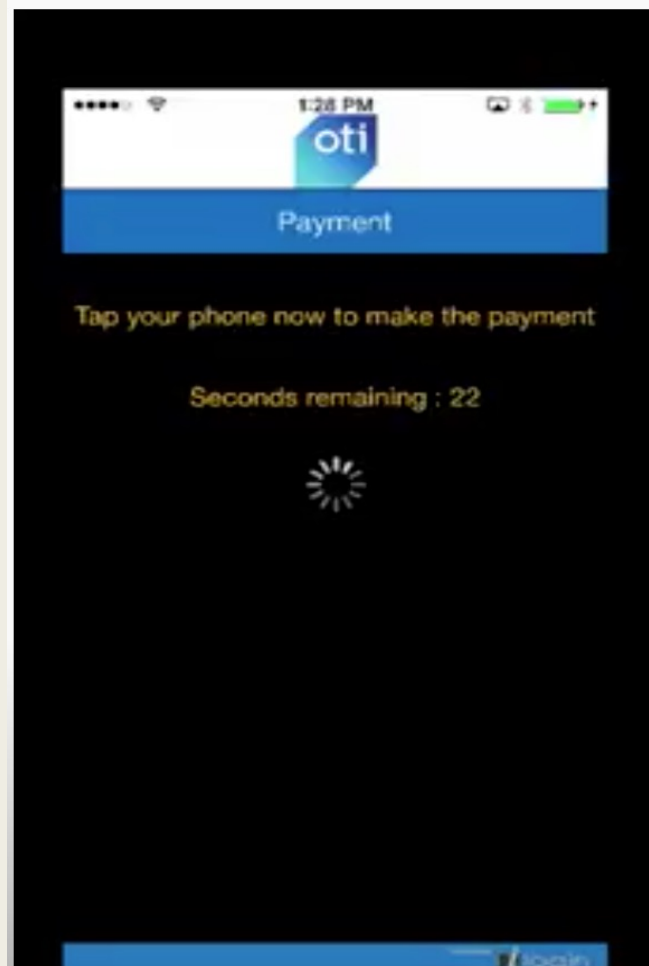
iLogin OTI App for NFC Phones: <https://www.youtube.com/watch?v=E7XGYJDnUn8>





Founder iLogin.com

iLogin OTI App for NFC Phones: <https://www.youtube.com/watch?v=E7XGYJDnUn8>





Toys”R”Us & Babies”R”Us

E-COMMERCE PLATFORM & WEBSITE
MOBILE APPS
INCREASE CONVERSION
PERSONALIZATION
ONLINE AND INSTORE BABY REGISTRY
AWS & IBM DATA CENTERS
SSO
SEO
LOYALTY
OMNICHANNEL REAL-TIME INVENTORY



Toys”R”Us and Babies”R”Us

Executive Leadership Council, Head of Engineering E-commerce

As Head of Digital Engineering and tech leadership team at Toys”R”Us leading 500+ FTEs and contractors, I was responsible for delivering e-commerce Omni-channel platform for Toys”R”Us, Babies”R”Us & Baby-Registry ecommerce website, mobile apps, micro-services platform, inventory management and Single-Sign-On (SSO). Before I joined, Toys”R”Us had outsourced their website to the GSI Commerce, now named as Radial. In order to improve the customer experience and operational efficiency of web sales, Toys”R”US started its in-house re- platforming project in the summer of 2015. After a couple of years of delay and burning close to \$100 million dollars with IT implementation partners, Toys”R”Us established their own digital engineering organization in late 2016 to launch the new website. I was hired to lead this effort. As of August 21, 2017, Toys”R”Us website and mobile apps was running 100% on the in-house platform handling peak holiday traffic.

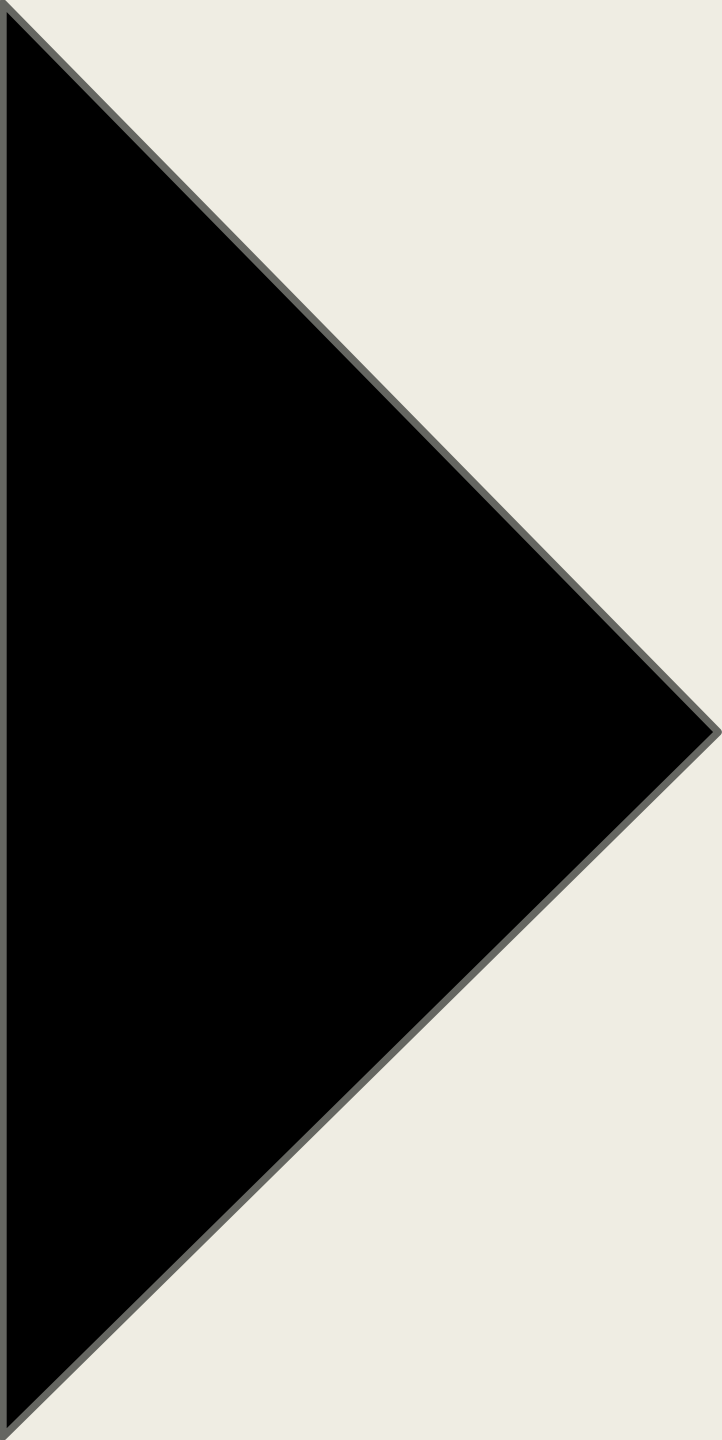
- Key member of the executive leadership council
- Head multi-billion-dollar Toys”R”Us, Babies”R”Us & Baby-Registry ecommerce website, omnichannel real-time inventory software, Single Sign On (SSO), mobile-apps, Loyalty, Search, Promo Engine, Personalization
- Achieved 4% increase in online conversion for the websites, Average Online Order Value by \$6, higher penetration of In Store pickup (ISPU) by 15%, 10% increase in Digital Baby Registry creates
- 2X gain in development and deployment velocity of the ecommerce platform by moving key functionalities from Oracle legacy ATG stack to React.js and micro-services and move to AWS.



Toys”R”Us and Babies”R”Us

Executive Leadership Council, Head of Engineering e-commerce

- Under my leadership we hosted 30 to 40% of the systems to AWS Cloud with features such as auto-scaling, multi-AZ HA, auto-region failover, and immutable infrastructure with instant rollback capabilities. The other 70 to 60% was hosted in IBM Cloud
- Thought leadership for Real-Time-Inventory software platform, a cross-functional initiative, to centralize inventory across Distribution Centers (DC), Fulfillment Centers (FCs) and stores. Introduced event-based architecture to integrate fragmented legacy systems. The system uses Kafka and Cassandra for event aggregation and elastic-search for faster access. This system powers website real-time allocation to prevent over-selling and reduce cancellation of hot items. It also leveraged to improve website conversion and introduce new features like dynamic bundling & dynamic pricing.
- Evangelist for DevOps and Agile at Toy”R”Us. Instrumental in establishing the dev-ops team that introduced automated build, testing, and deployment using Git and Jenkins to the software development lifecycle.
- Thought leader and technology sponsor for SSO (Single Sign On) a much-needed platform to improve customer experience by bringing fragmented credentials/authentication for baby-registry, loyalty, and e-commerce functions. Under my guidance, the team developed a custom-built federation engine with API interfaces and multi-factor authentication capabilities.



ELF OBSERVABILITY FRAMEWORK

ELF Observability Framework

Conceived the idea and Led team to deliver ELF Observability Framework

ELF was developed to reduce MTTD (Mean Time To Detect) and MTTR (Mean Time To Repair) issues in end-to-end application journeys in production by addressing the issue related to fragmented log collection and monitoring systems. ELF is built using OpenTracing and OpenTelemetry standards. ELF provides the ability to:

- Centralize data sink to capture application raw log data for all applications
- Trace user journeys end-to-end. A trace is Directed Acyclic Graph (DAG) of Spans. Implicitly weave in millions of existing logs at Amex into OpenTracing traces/DAGs.
- Visualize and debug application trace and log data using distributed open tracing standard
- Baseline, Monitor and Alert on anomalies, spikes and other patterns of interest (with anomaly detection)
- Ensure a federated consumption model where applications can tap into the data sink and consume trace, span and log data

ELF is architected, designed and developed to be horizontal scalable using open-source products. It uses Amazon's Open Distro, Apache Kafka, FluentD agent, Jaegar UI (integrated with IIQ) and ElasAlert as part of the technology stack

C++ and JS ELF library inner sourced and co-developed with CAS and Customer Acquisitions teams.

ELF Observability Framework

ELF Accomplishments

Apps in Production

~430 apps

1500 oneData functions

Apps in Dev and QA

~150

Average events/day

~5 bn

Average traces ingestion/day

~285 million

Average spans ingestion/day

~1.5 bn

Average log ingestion/day

~3.5 bn

Average data ingestion/day

4.5 TB

Peak ingestion:

200 MB/sec



ELF Observability Framework

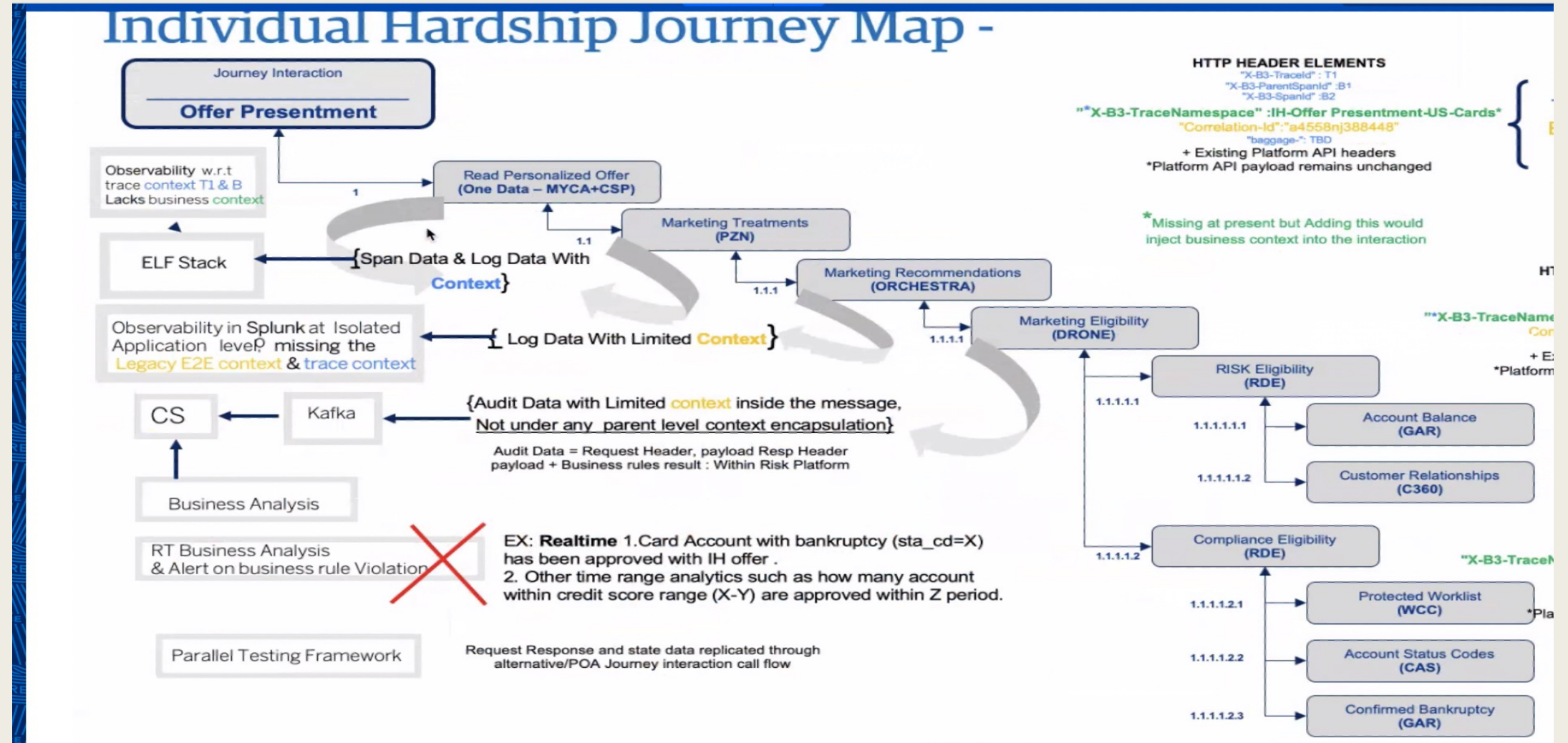
ELF Features

- Built on OpenTracing standards. It also implicitly weaves in existing millions of log4j & logback logs at Amex with OpenTracing Traces and Spans
- OpenTracing OpenTelemetry Alpha ELF collector released
- End-to-end dynamic tracing and logging of application journeys at Amex
- End-to-end correlation of application logs across application journeys
- Modified Jaegar UI to integrate with Amex IIQ and visualize logs weaved in with traces and spans DAGs (Directed Acyclic Graphs)
- Highly optimized Query Server to return search results in sub-seconds
- Baseline, Monitor and Alert using open-source ElasAlert for alerting on anomalies, spikes and other patterns of interest(with anomaly detection)
- Track top N slow end-to-end ELF traces & logs
- Uses Random Cut Forest ML algorithms to detect anomalies and reduce false positive alerts by detecting seasonality and patterns on attributes being monitored
- integrated with Sahara to generate ServiceNow tickets, slack, Jira tickets and email

ELF Observability Framework

Sample ELF Journey – Individual Hardship Journey

(Offer Presentment-> Read personalized Offer -->PZN (Marketing Treatment) --> Orchestra (Marketing Treatment) -> Drone (marketing Eligibility) -> RDE (Risk Assessment)





CAS – ENHANCED STANDIN (ESI)

CAS Enhanced Standin (ESI)

Principal Architect, CAS Enhanced Standin

Enhanced Standin stands in for Authorization when TPF is down. This helps make better auth decisions when CAS TPF is down. ESI is not a replacement for Nemo Standin. Nemo still stands in for any auth requests that time out in TPF or ESI.

ESI is an example of how two teams (CAS Delivery and EA teams) work together as one team. With help of 4 Staff Architects embedded into CAS, the combined Enhanced team architected, designed, developed and delivered CAS Enhanced Standin in 1 year

- ESI stood in 11 times in 2020 processing 500K transactions at \$53M
- Quality risk assessment on high spend transactions vs. 100% declines in Network Stand-in
- Fraud Risk: 10% increase in stand-in approvals with 50% lower fraud rate
- Credit Risk: 6% increase in stand-in approvals for customers with less risky, better risk profiles

Enhanced Standin helps make better decisions using dynamic credit and fraud risk rules. Another benefit of ESI is we are now able to provide authorizations decisions for transactions more than a certain dollar value, during Stand-in in regulatory markets such as Singapore which were hard declines prior to ESI.

CAS Enhanced Standin (ESI)

Principal Architect, CAS Enhanced Standin

Enhanced Stand-in (ESI) created the first authorizations engine built with modern distributed technology stack at American Express, which ensures the safety and soundness of improved risk decisioning while offering a better customer experience by minimizing disruptions at point of sale in the event of a disruption with Credit Authorizations System (CAS).

ESI created the first of its kind high velocity data pipeline with High-Speed Connector and Kafka to synchronize and distribute data from the TPF mainframe to the distributed platform in real time.

The architectural needs of this program led to the creation of multiple EA Frameworks that are used today:

- EA Unblock (orchestration to avoid writing complex, asynchronous reactive programs)
- Source-Sink (moves data from source to destination – i.e., TPF to Kafka, Kafka to Oracle...)
- Inner sourcing of C++ Enterprise Logging Framework Libraries that was used for observability and anomaly detection
- GCAG-ISO Java Parser library

Additionally, ESI was instrumental in enhancing the Rules Engine to be scalable and performant while processing a high volume of transactions.

This program created:

- New platform performing Credit and Fraud risk assessment when the Network platform detects a degradation in CAS responsiveness
- Significantly enhanced customer experience for card members with high value transactions as well as card members in regulatory markets without increasing risk to the company
- First building block for the Authorizations Modernization Program on modern architecture
- Modern CI/CD Pipeline with parallel automated testing, static analysis, and a rigorous code review process



LUCY RULES EXECUTOR RULES PROPAGATION

Lucy Rules Engine Executor

In addition to delivering Enhanced CAS Standin with CAS delivery team, I led the EA team to work very closely with Lucy Rules Engine Team to enhance the Lucy Executor and rules propagation to CAS Enhanced Standin.

While working with CAS team for AMP POA, I continue to lead the EA team to architect and design rules propagation system for Lucy Executor across regions. We're also ensuring what is delivered for Lucy Rules Engine are the key building blocks for ensuring American Express has one Risk Rules Execution Framework including the creation of "Lucy in a box".

Some of the ways we're doing this are through ensuring the following requirements are met:

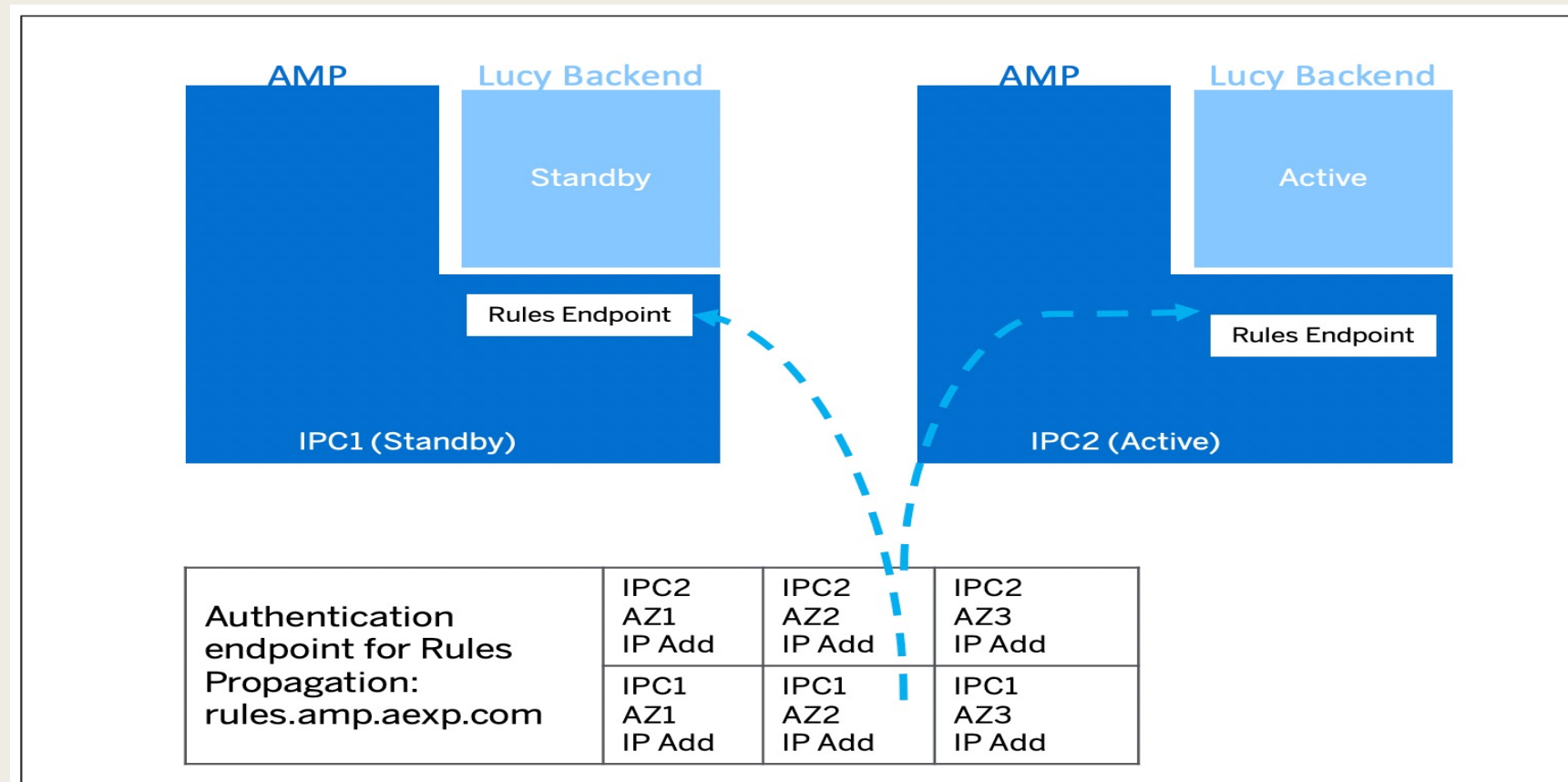
- Highly available, performant and secure
- Scalable
- Strong consistency

The above are now part of standard Lucy Rules Engine.

Lucy Rules Propagation

Principles for Rule Propagation:

1. Rules across pods in IPC2 and IPC1 must be coherent (Unless there is a manual override)
2. Failover of Authentication compute is transparent to Lucy Backend



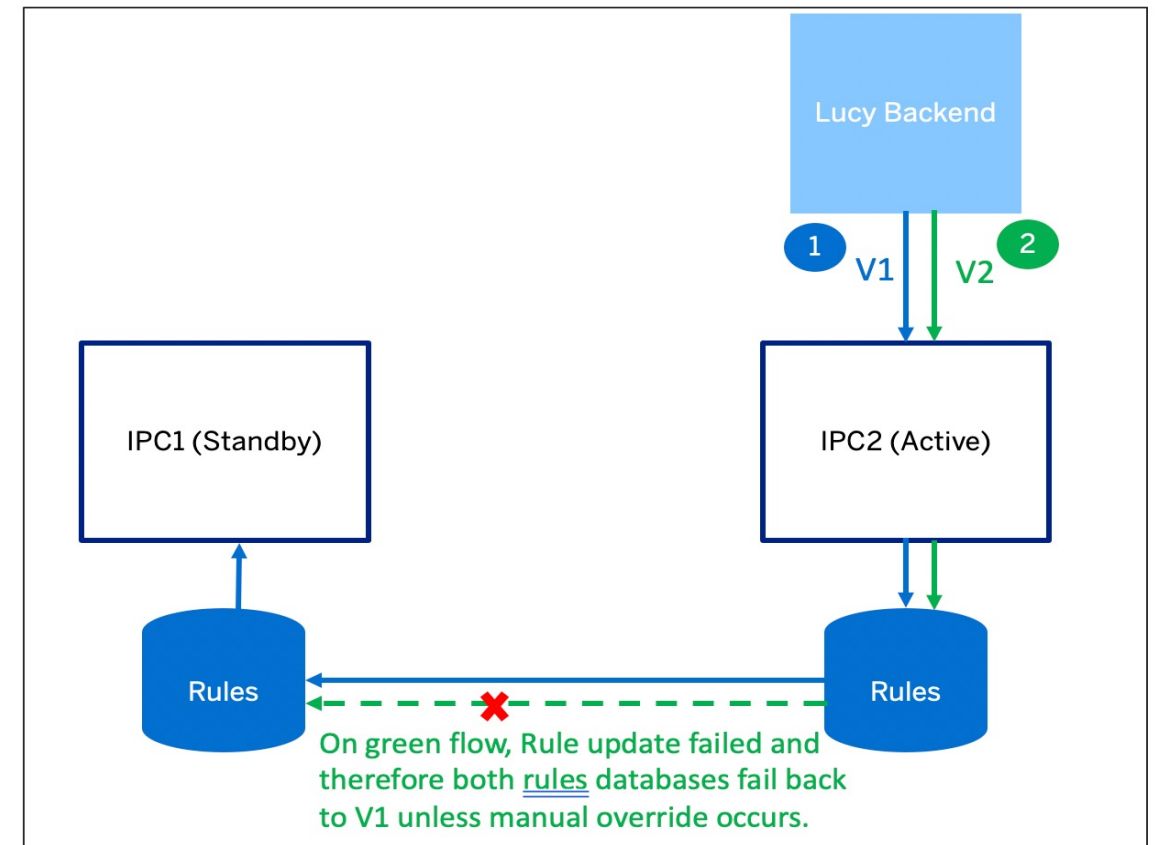
AZ1, AZ2 & AZ3 are the 3 Availability Zones in each region. Rules Endpoint is rules.am.aexp.com

Lucy Rules Propagation

Principles for Rule Propagation:

1. Rules across pods in IPC2 and IPC1 must be coherent (Unless there is a manual override)
2. Failover of Authentication compute is transparent to Lucy Backend

Commentary	Lucy Backend	IPC1	IPC 2
	1 V1	V1	V1
Update failed to <u>IPC2</u> so both revert to V1	2 V2	V1 V1	V2 V1
Manual Override	V2	V1	V2
IPC2 Failed over to IPC1			
Rules Diverged Manual Override	V2	V1	V2
Heartbeat between Lucy backend and IPC1 detects divergences and updates IPC1 to V2	V2	V2	V2



V2 V1 V2 Rule snapshot versions



AMP POA & RDM 2.0

AMP POA and RDM 2.0

Principal Architect, AMP and RDM 2.0

AMP POA and RDM2.0 is an example of how two teams (Risk Decision Tech Delivery and EA teams) work together as one team. With help of 4 Staff Architects embedded into Risk Decision Tech, the combined team is Architecting, designing and building Authentication Platform as an MVP

AMP Authentication is the first use case being built to utilize AMP POA and RDM2.0. The purpose of AMP Authentication is completing a risk assessment for login from MYCA to determine if 2 Form Authentication needs to be used for the MYCA login

AMP POA and RDM 2.0

AMP Platform design objectives

I used the following objectives as a guiding principle to Architect, design, development and deployment of AMP POA and RDM 2.0

- Network in a Box: Prop issuer treated like any other issuer by Nemo
- Flexible and extensible to support risk assessment for future products including
 - Debit
 - Push Payments
 - Direct to Bank
- Reduce time to market for new features
- Support Data Localization
- Consolidated Risk Data Platform across Authorization and Underwriting

AMP POA and RDM 2.0

Principal Architect, AMP POA & RDM 2.0

- Architect and design AMP POA & RDM 2.0 to replace CAS TPF & RDM 1.0
- AMP POA and RDM 2.0 are designed to be highly scalable, config driven and performant.
- GNS Fraud Risk Assessment designed to be collocated with Nemo Cloud
- Unlike CAS TPF, GNS Authorization requests architected to not cross continents. This will help improve SLA for GNS auth transactions
- RDM 2.0 designed and abstracted from AMP compute. AMP compute does not have direct access to the AMP RDM 2.0 Store. The design provides a CRUD interface to allow for new types of databases for RDM as required in future
- RDM 2.0 provides a materialized view for Risk Variables
- Architect and design AMP multi-AZ HA
- AMP failover to another region after 2 AZs in any region are down
- AMP designed to have HOME for accounts. Some accounts are Active in IPC1(say) and some in IPC2(say). This allows both data centers taking traffic and not create an element of surprise when doing failover from one region to another
- AMP core designed to use Tokens and no clear text CM15, CMM11



Enterprise Data Vault (EDV)

In E3 Production



Enterprise Data Vault (EDV)

Platform to tokenize clear text PII data



- EDV is a Key, Value Store where Key is the PII Data, and the value is the token or vice-versa
- EDV is a Hub and Spoke Architecture with Spoke in each Region or co-located with SORs that cannot tolerate latency
- Clients request for tokens for one or more clear text PII data using EDV API
- EDV API tries to find the token in the EDV L1 LRU cache (EDV L1 LRU cache cache is in-memory cache)
- EDV API when not finding a token in EDV L1 LRU cache, tries to find the token in EDV L2 cache (EDV L2 cache is the EDV satellite DB)
- EDV API when not finding the token in EDV L2 cache in the EDV satellite, tries to find the token in EDV HUB
If the token for this PII data is not found EDV creates a token for this PII data
EDV Hub then propagates the token to all the EDV satellites
- EDV Hub will be pre hydrated with PII Data and Token mapping.

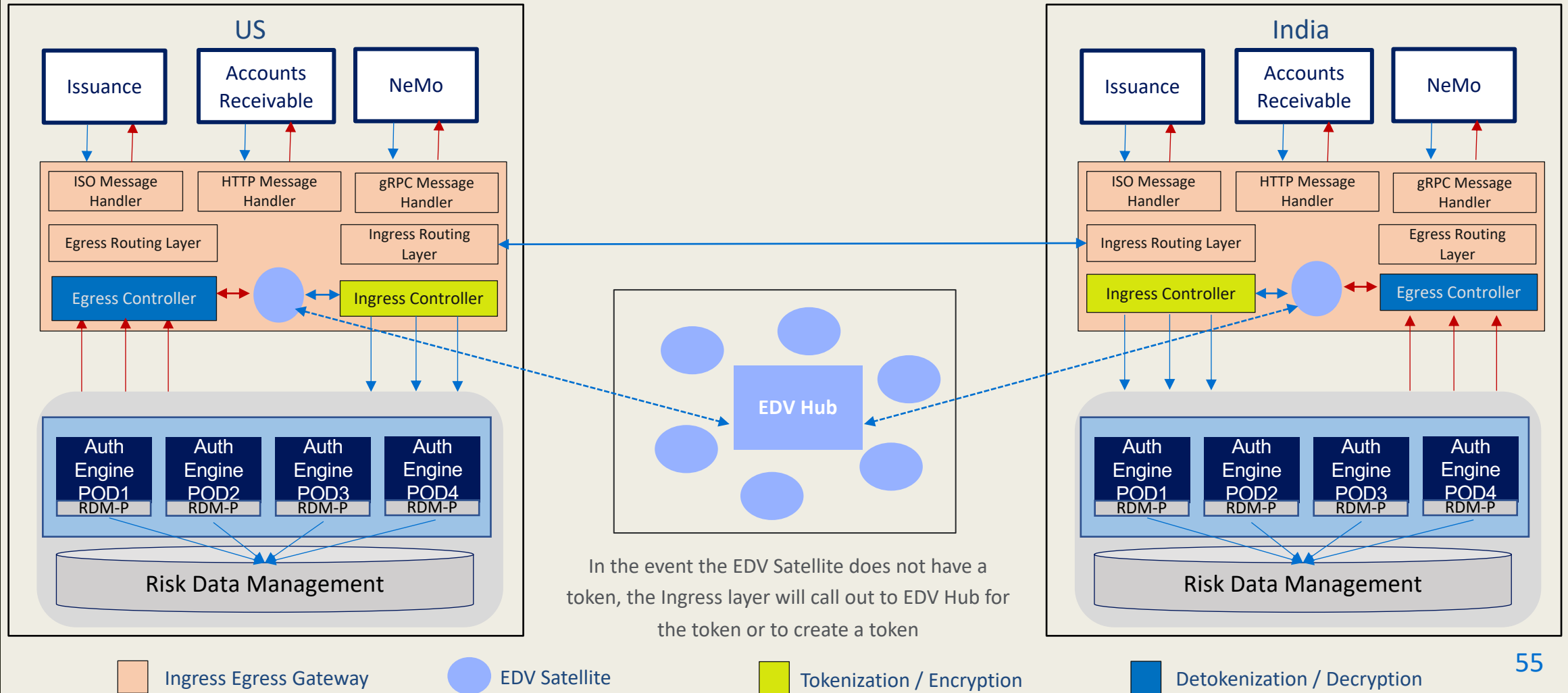


Enterprise Data Vault (EDV)

Platform to tokenize clear text PII data



The picture below shows an EDV Satellite co-located in the Ingress Egress Gateway of the SOR





Ingress & Egress Gateway

In development

Ingress and Egress Gateway In development

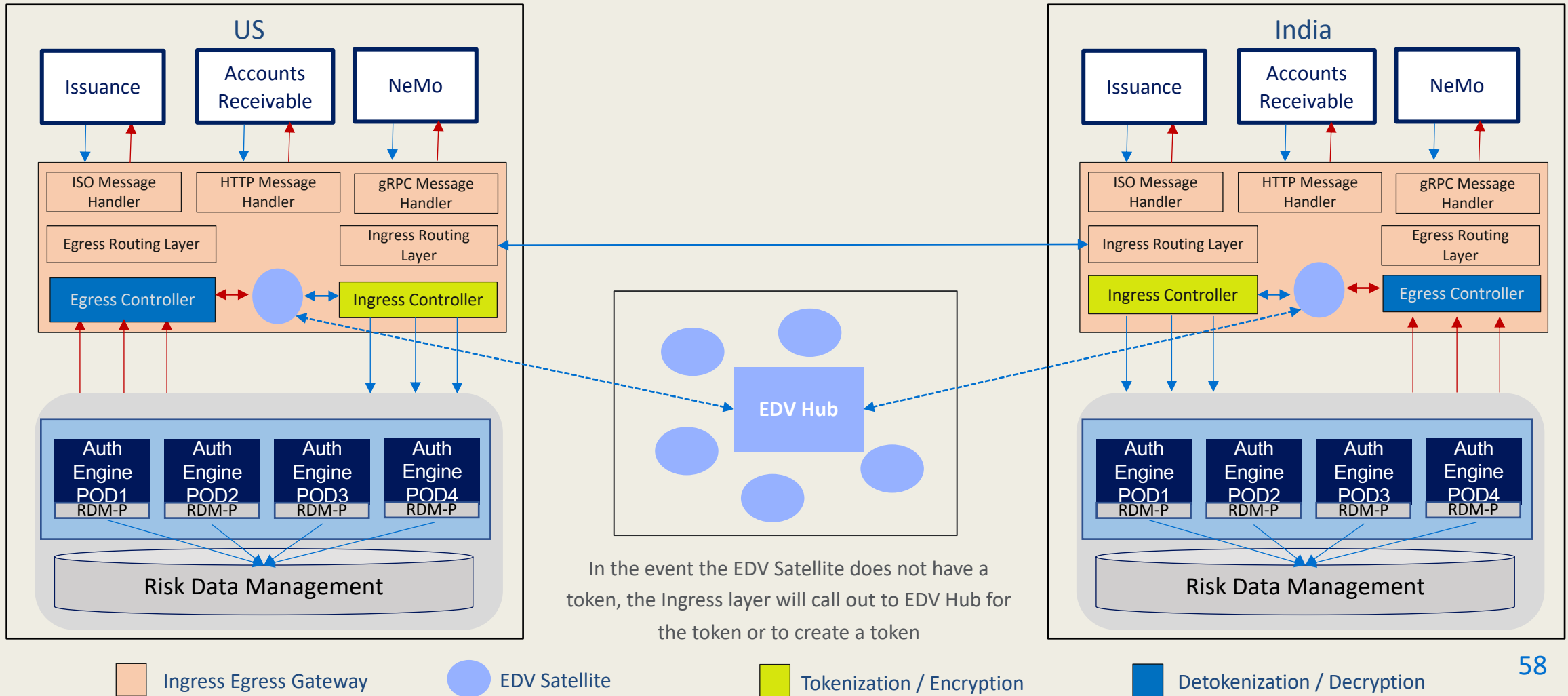
Gateway to route messages for Data Localization & tokenize any clear text PII data

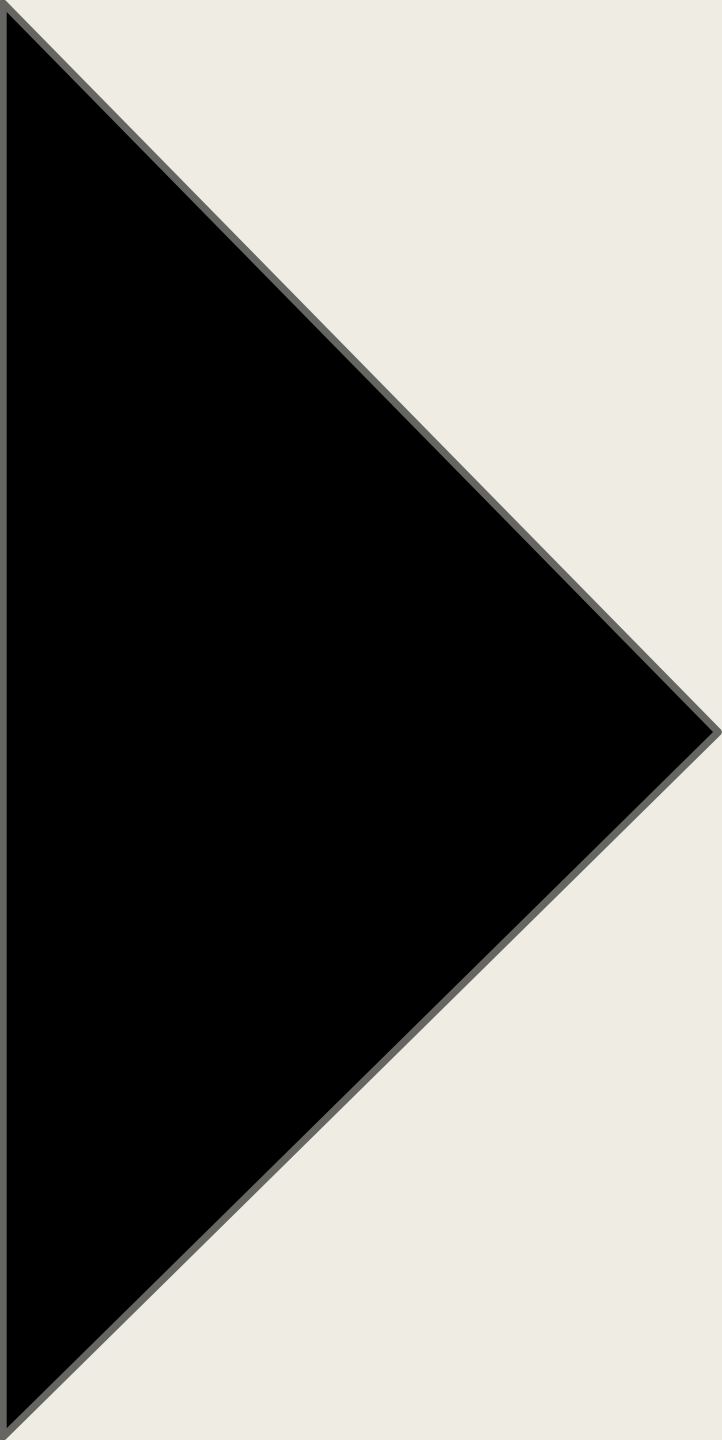
- Ingress Layer will route messages and tokenize or encrypts clear text data for all incoming sensitive data elements if the consuming SOR is designed to use Tokens
- Egress Layer will detokenize or decrypt all sensitive data elements that are sent to other SORs that do not understand tokens
- Enterprise Data Vault (EDV) will be used by AMP for Tokenization of sensitive account data elements primarily through EDV satellites co-located with AMP

Ingress Egress Gateway In development

Gateway to route messages for Data Localization & tokenize any clear text PII data

The picture below shows the Ingress and Egress Gateway in US and India. This gateway routes requests to their HOME region and tokenizes the clear text PII data if needed





EA-SOURCE SINK

EA-UNBLOCK

ISO MESSAGE PARSER

EA-UUID

EA Source Sink, EA Unblock, and ISO Message Parser

As mentioned, in delivering Enhanced CAS Standin with CAS delivery team, I led the EA team to create

- EA Source Sink Framework
- EA Unblock Orchestration Framework
- ISO Message Parser to Java POJOs (yet to be extracted out as a framework)

These frameworks were added to the EA Frameworks portfolio and can now be leveraged by other teams at Amex

EA Source Sink Framework – Allows to plug in Source and Sink to move messages from a given Source to Sink. In ESI this framework allows for moving messages from TPF High Speed Connector to Kafka, Message Receiver receiving messages from Nemo GAN to Kafka, Message from Kafka to Database in ESI Orchestrator

EA Unblock Orchestration Framework – Allows programmers to write complex multithreaded reactive code as linear programs

ISO Message Parser to Java POJOs – This library parses ISO messages and generates POJOs

EA-UUID

High performant statically unique Ids in space and time

In addition to delivering ELF, I led the EA team to design build EAUUID Id Generator in Java and JavaScript. EAUUID are statically unique Ids in space and time. These EA-UUID are 3X faster than Java GUIDS. These are used as ELF Traceld and ELF SpanId

This library was added to the EA Frameworks portfolio and can now be leveraged by other teams at Amex



STATIC CODE ANALYSIS

CODE FORMATTERS

CODE REVIEW CHECKLIST



Static Code Analysis, Code Formatters & Code Review Checklist



Championing the adoption of the following static code analysis tools, code formatters and code review checklist

For static code analysis for ELF, CAS/ESI, Lucy rules Engine CI/CD pipelines we use the following tools for static code analysis.

- pmd: for java in CAS/ESI/ELF/EA projects
- Detek: for kotlin in EA/ELF projects
- Eslint: for javascript/typescript in EA/ELF projects

Code Auto formatters used are:

- ts/js: prettier.js
- java: spotless
- Kotlin: klint
- Golang: built-in

Formalize code review checklist:

<https://enterprise-confluence.aexp.com/confluence/display/ELF/Pre-review+checklist>



AMEX HYBRID CLOUD DATA STRATEGY

Amex Hybrid Cloud Data Strategy



Working with Dr. Man Chan U (Kevin) and other stake holders to produce Amex Hybrid Cloud Data Strategy