#AboveAndBeyond, #SimplifedInvented #CustomerCare #CollaborateTeam #Ownership #HighStandards

**CRA - Compliance Factory CoR**

* AMR team: Due to amalgamation compliant checks had become tough to maintain
* Customer docs can belong to multiple accounts, each account have diff checks (checks can be common).
* Handle changes in on/off for account, order of checks, commonalities, cycles.
* Collaborated with business team: all that info we got from
* DynamoDB configuration file to store changes.
* Factory spit out the chain.
* Each chain is compliance.

Numbers:

* 10-12 requests for change /month by ministries.
* 5-10 documents uploaded by customers per month
* 40% of checks were common.
* Reduced processing time by 30% (from 10-12 sec to 6-7 secs)
* Deployment went from week to 1.5 day.
* 95% code coverage, using JaCoCO

#Conflict #Customer #DiveDeep #highStandards #EarnTrust #TightDeadline

**CRA - Distributed Transactions Refund Disbursement and Correspondence**

* SOA team: RMS, DS, and CS were three services, recently split, needed to maintain consistency.
* Customer were getting inconsistent updates, even after 48 hours, (5% misclassification).
* Colleague proposed 2Phase commit (too slow), No direct db access, but through APIs.
* DS had distr Locks to handle funds
* Throughput of RMS and DS reduced, Strong consistency
* I suggest, even driven arch, Kafka, Spark Stream(500-100 collected and then sent to DS) (reduced Dist locking), Flink (eventual const.)
* Understood colleague. Familiarity with 2PC, balanced pros and cons.
* Idempotency to remove delicacy.

Numbers:

* Got 2M fund reqs, completed all, before only could <1M.
* Consistent msg to customer, (RMS response time 1 Day to <10 secs).
* Reduced dist locking 1/500.

#Customer #DiveDeep #Invent #Action #HighStandards #Mistake #Deadline

**CRA - FFD File Categorization**

* FFD team: Meeting with dispute service team; 65% of tickets to transfer tickets. ("audit adjustment" or "late filing").
* Reduce misclassification, read by OCR, reduce manual dependency.
* OpenNLP – Naïve Bayes: tokenization, classification (72% acc). NO Contextual understanding. No multi label.
* Pre-trained BERT model with Amazon SageMaker: Context underst, multi-label. (85% acc)
* Tesseract OCR: for images
* Real acc: 80%, reduced misclassification: 65 – 20%., reduced manual efforts.

#Backbone #NegativeFeedback

**CRA - Negative feedback**

* FFD file categorization story: focused a bit too much on accuracy.
* By passed an important detail, where few categories were interchangeable (same team).
* Changed model, increased accuracy to 91%

#Backbone #Concede #Mistake

**CRA - Optimizing Dispute Assignment and Scheduling with SQS – Concede**

* FFD team: During peak seasons, got many complaints (20% complaints) simple disputes took >week.
* Some cases complex tasks hogged time and workers.
* Improve load dist algo, ensure scalability, simple integration.
* I proposed: Multi Queue scheduling: 70-20-10 dist. (FFD throughput decrease have to calc complexity)
* Real issued lied in backlog DB, cos most new disputes went to DB not directly to workers.
* Colleague proposed: modifying dist from backlog DB (easy integration, index by complexity and time)
* Loaded cache every day with 70-20-10 of new disputes.

Numbers:

* Off-Peak Periods: 50,000 disputes/week.
* Peak Tax Seasons: ~70,000 disputes/week.
* Low Complexity (50%) 10hr to 3hr
* Medium Complexity (30%) 3 Days to 1 Day
* High Complexity (20%): Week

#DiffPersonality

**CRA - Different Personality**

* FFD Team: SQS scheduling: I – semi formal vs Coll: ad-hoc, dynamic
* Documentation vs No documentation
* Began and ended meetings on light note: Middle: documenting key decisions pros and cons.
* Conducted barnstorming sessions: ad-hoc and free flowing to fulfil need and think out of box

#NewFromOutside #DisagreedManage #HighStand #FrustCustomer #Invent #ThinkBig #Mentor # BckCompat

**MW - Improving Computational Efficiency in a Multi-Rate Simulation System**

* Simulink: EV manufacturer speed and braking system diff rates (10ms and 1ms).(50subsyst – 1000models)
* 80-90% CPU cycles wasted, low efficiency, more battery.
* Frustrated customer; cant replicate actual working; Diff in real results.
* Increase efficiency, easy integration reduce to 50% cycles wasted.
* Team thought of changing block or modifying current block to adjust the rate (too much effort for customer).
* Workaround: Customize MLFB script, which I wrote.
* Used JProfiler to pin-point inefficiencies Brought event driven web arch from diff team; Publish-subscribe pattern to propagate events.
* Presented to stakeholders with backward compatibility, only config change to push than pull.

#FrustCustomer #BckCompat #Future #LongTerm #HighStandards

**MW - MLFB Dimension Issue**

* MLFB Team: Automation indust. MFLB sporadically converted row vector to scalar.
* Occurred in about 2-3% services, where more than 100 MLFB ran simul.
* Assured that issues is not theirs, gave them work around of script to maintain consistency.
* Reached higher ups to own project; long term fix took 2 months to debug.
* Under memory constraints (Optimization routine: background process) MLFB compressed same number row vector to scalar. And due to exception, prev value was thrown.
* Changed arch. To introduce strict mode for vectors (No change).
* Defined global method to get vectors even if exception; method de-compressed vectors. (To make sure it doesn’t appear again)

#FrustCustomer #OverCommitted #DeepDive #Mistake #Deadline

**MW - Overcommitting on Optimizing MATLAB Backend Performance for Large Data File**

* DataJoint: Neruoscience project processed and uploaded file using DataJoint pipeline, couldn’t do it for 10GB fil, threw exception.
* Workaround: broke file into parts manually took 2 hours to process.
* Explained Customer limitations (5GB; cos memory loading) overcommitted that would reduce time to 45mins due to multi threading. (DataJoint didn’t support threads)
* Serialization overhead: MAT-file serialization: interm states; endianness data
* New Algo: Essential Data, Binary Encoding
* Compression+splicing: reduced to 30%

Data: EEG Signals

* Profile info
* Reaction times
* Reaction patterns
* Metadata

#NoData #QuickThinking #BiasAction

**MW - Quick Decision**

* Matlab coder baseline generation: Tool to upload models, stored in S3, service pulled models, ran MATLAB native fcn, generated baseline and stored in S3.
* Attempted to centralize the baselines, cos multiple teams shared features but maintained diff baselines.
* Last stage test cycle, some tests failed for baselines sporadically ~5%; Quickly realised that it happened for models with 50+ nested hierarchy.
* Changed algo in Java to store only immediate parent, made it default generation algo and fixed the issue.

#ShotDown #Management

**MW - Simulink Inheritance**

* Each release will get 5-10% client request to change config, behaviour of Simulink service feature (cos they would encounter new cases and would want us to resolve it beforehand).
* Borrowed concept from JDBC plug N play; Expose interface to customers.
* Built prototype; POC for manager and upper management. And business.
* Got shot down due to security reasons.
* Would continue to find ways to isolate customers.

#Invent #Ownership #Scratch

**MW – Employee Metrics**

* Existing system for managerial metric was outdated, only displayed raw numbers.
* Proposed full-time project hackathon; backend using Spring Boot for scalability and API integration.
* Used HBase analytics capabilities and efficient time-series data queries.
* Fetched employee metrics weekly from APIs through scheduled Spring Boot jobs
* frontend Angular, Chart.js for interactive graphs and NgRx for state management.
* Delivered role-based views for managers and employees with advanced search and filter options.
* Achieved 85% coverage. Increased managerial adoption by 50% due to improved usability and actionable insights.

#DiveDeep #HighStandards #ScalableSolution #CustomerObsessed #CostEfficiency

**CRA - Account Transaction Query Optimization:**

* Account transaction: Diff team/Clients were making analytical queries over historical data.
* Even after time indexing, it was difficult as still need very long table scan, cos of user ID.
* Wanted to improve query performance, ensure strong consis; ACID trxn.
* While large, the volume of historical data was manageable
* Sharded data by year (query patterns: New relic): used materialized views; Cached frequently access data.
* Reduced response time by 60%.

**Learnt JAX:**

* SOA: After consolidation of AMR, called me in for this. Diff stack.
* Learnt JAX: GET, POST, Path., filters and interceptors for authentication and logging.
* Reduced redundant calls by 90%.

**MW – Sent away customers**

# Licensed Figure Manipulation Tool Restriction

* **Challenge**: A licensed MATLAB figure manipulation tool was non-deployable due to licensing restrictions, preventing interactive plot adjustments in the compiled application.
* **Solution**: Built custom controls using uicontrol for zooming, panning, and scaling, providing sufficient interactivity and ensuring future compatibility.

**Tell me about yourself:**

* Problem solving has always been a passion, loved puzzles as a kid.
* Early high school, learnt Java, Android Studio.
* Passion led me to Bachelors; laid my foundation; 3rd got placed in MathWorks.
* MathWorks : Mostly Focussed on backend; worked closely with customers esp 6 months.
* Understood business, large codebases.
* Contr: Led and delivered projects with major arch changes, resolving dim accuracy.
* Worked there for 4.5 year; Never stopped learning -Cloud, Dist, System Design
* Curiosity led me to masters. Studies in hectic env with dynamic group of people. Learnt a lot; last sem got
* CRA: Tackled large-scale system challenges; Owned projects, Led and delivered major addition to features FFD, AMR, SOA, AccountTrans. Reduced technical debt, code smells; Optimized backend services keeping them scalable and resilient.

**Why Company:**

* College mates; 2 friends working at AWS currently, their experiences have been inspiring.
* Trusted with ownership of service early on; Given full control of products, freedom, warehouses visits.
* Seen products go live within weeks, making millions;
* Seems like a company, everyone thrives to build next big thing.
* Values and reward passion and skillset.
* Ideal for a geek like me where both professional and personal life can become better.
* AWS itself went from internal tool to leading cloud techs
* CP: The Aurora Control Plane team focuses on building scalable services and tools to support this rapid growth, ensuring reliability and operational effectiveness.
* I personally resonates with this; always push for new ideas, built raw kitchen app; nobody asked for.
* Worked closely with business/customers; led and delivered end-end prdocuts; customer expectations.
* Ample time reading AWS whitepapers; gaining insight behind thought process behind products architecture, seems like an env where solving challenging problem is a norm
* Beyond technical aspects: Social values, pandemic and floods reinforced my belief that company is making a positive impact on society

Questions to ask:

* What excites you the most about Aurora's roadmap and the Control Plane's role in shaping its future?
* Aurora’s pace of innovation is remarkable, I have seen teams making this trade-off, how does the Aurora CP balance the introduction of cutting-edge features with maintaining and optimizing existing ones?
* AWS is leading cloud platforms and Aurora is a high-performance cloud database service. From an operational perspective, how does the Control Plane team ensure that Aurora maintains its edge over competitors DB services?
* Every system has moments of downtime or incidents. How does the Aurora Control Plane team incorporate learnings from such events to improve operational resilience and customer trust?
* How Aurora Control Plane team, encourages and support engineers in experimenting with new ideas.
* How is feedback shared within the team—both for personal growth and project improvements?
* What metrics are most critical to the success of the Aurora Control Plane
* AWS is known for providing opportunities to learn and grow.
* What are the key technologies and frameworks used by the Aurora Control Plane team,
* What are the current technical challenges the Aurora Control Plane team is working on