

SRECon Singapore: June 2023

From Keeping The Lights On
To Designing The LEDs



standard
chartered

Introductions & Level Set

Who are we?

Why are we here?

Why SRE?



Introductions: A bit about us



Sriram Subramanian a.k.a Sri

Managing Director,
Global Head of SRE & Engineering
Excellence, CCIB

Hobbies: World Music, Cricket,
Travelling and visiting UNESCO sites,
flying kites



Ian David Hamilton

Executive Director,
Head of Platform Resiliency and
Experience, CCIB

Hobbies: Spending time with family,
Cricket, Football, Cycling, Gym,
Blockchain



Introductions: A bit about Standard Chartered

83,000

We employ 83,000 people around the world



131

Our colleagues come from 131 different countries

160+

More than 160 years in business

653

653 branches worldwide

59

We are present in 59 markets

World's Best Employers

2022 Forbes List



100

Among the top 100 largest companies listed on the London Stock Exchange



2

We're listed on the London and Hong Kong Stock Exchanges.

- ≠ We were traditionally a trade finance bank and have evolved over decades to offer full set of banking services across retail and corporate customers.
- ≠ We are one of the trusted international banks with independent digital banking licenses SG and HK with names of Trust and Mox respectively.
- ≠ We also have a Ventures business with a portfolio of **22** companies and **30+** Ventures




Organizational Structure – Where we sit

How we are organised

Our client segments


Corporate, Commercial and Institutional Banking

Serving over 5,000 large corporations, governments, banks and investors.




Consumer, Private and Business Banking

Serving small-to-medium sized businesses and individuals, from massmarket to affluent and high-net-worth clients.

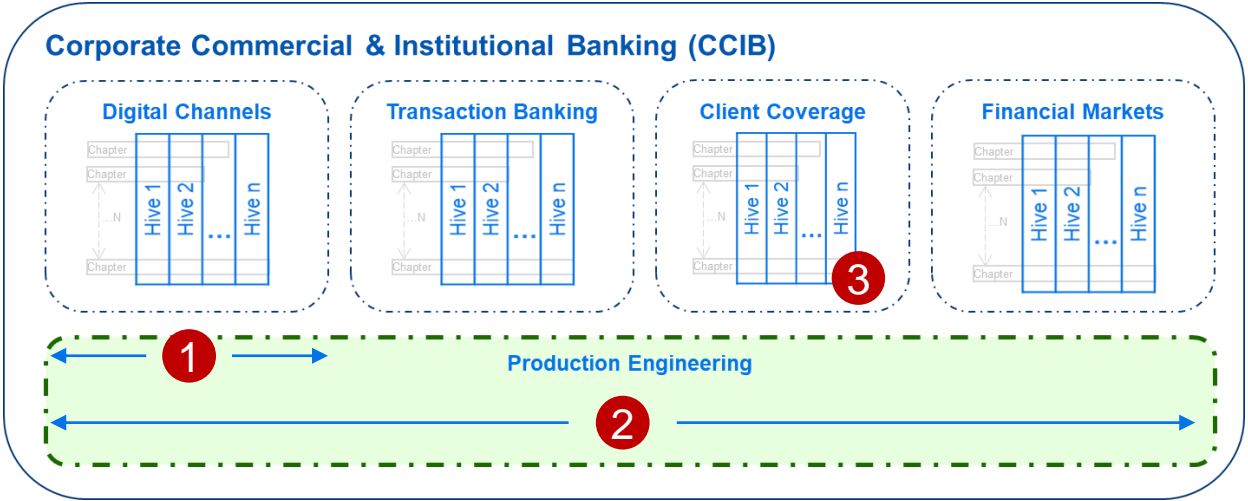


Ventures

Creating alternative business models in financial services and investing in disruptive financial technology.



CCIB



CCIB is a key business revenue generator of the bank.

Transaction Banking is evolving rapidly

Client Experience is paramount to growing transaction volumes

Three flavours of SRE exist:

- 1** Business aligned
- 2** Tools (Centralized)
- 3** Embedded



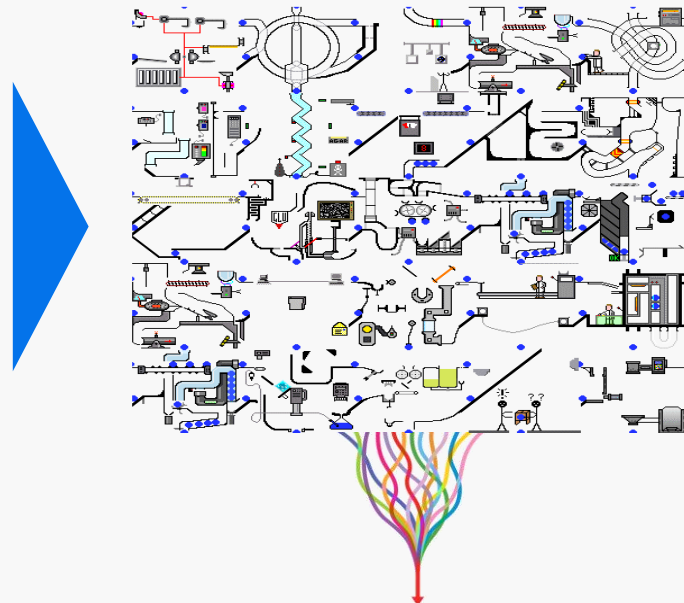
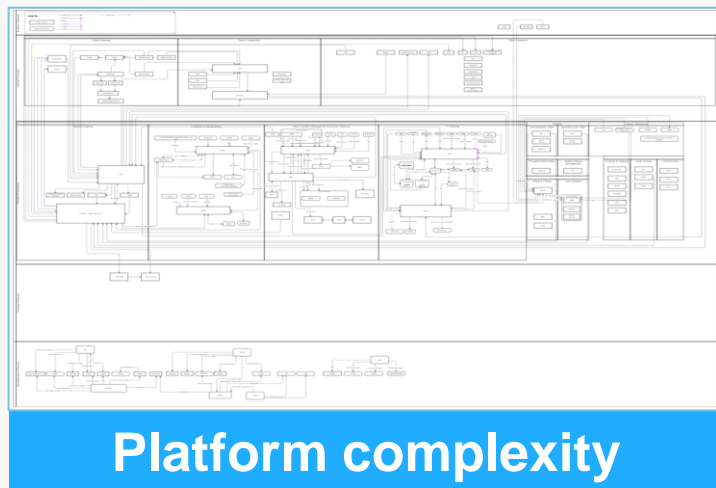
Why are we here?

- Executing a major SRE transformation, just completed 3rd Year
- 500+ Engineers, 100+ applications distributed globally
- Share our story
- Provide useful insights, focusing on the implementation of SRE
- And welcome some crowd sourced solutions

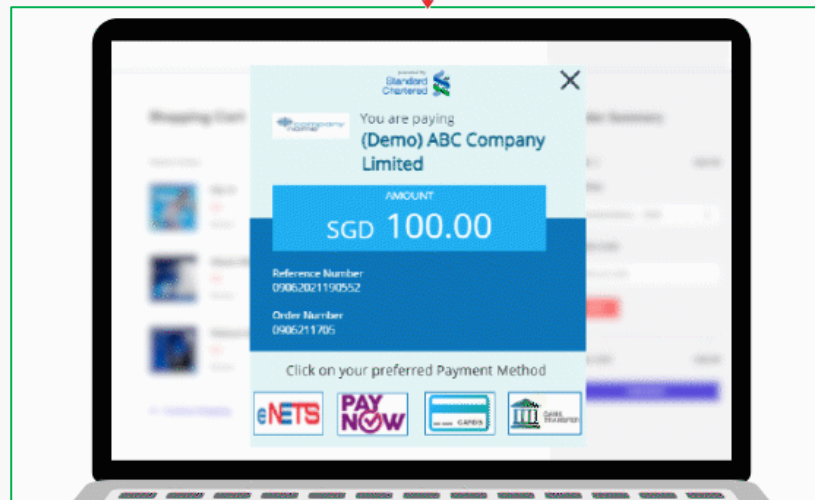
Key takeaways can be found here throughout our talk



Why Was A Transformation Required?



Customers expect us to be
"Always Aware" with
"Always On" services



Digitization is driving rapid creation of digital products as they provide significant revenue opportunity

Speed to Market to deliver these product capabilities at times **compromises F2B design thinking** and E2E service capability, resulting in Manual toil



Added Relevance: Society and Cashless Payments

Growing dependency has led to elevated regulatory oversight

“ After another bank outage, is it time to make banks publicly report service uptime? ”

Prominent Singapore Newspaper, Q1 2023

Regulators are likely to **gamify Reliability** to promote competition in financial services, with **banks asked to publish reliability statistics** as is done with internet broadband providers.



Solution: Why System Reliability Engineering (SRE)?



Inspired after attending the
Oct 2019 SRECon in
Dublin, we embarked on
our own SRE
transformation



What will we be sharing

1. SRE Tenets

2. Focus areas of our Transformation

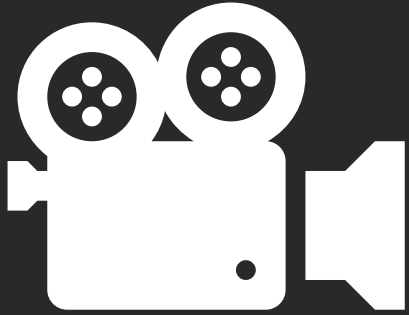
- People
- Process
- Tools

3. Wrap Up – Key Takeaways

Key aspects to be covered

- How has it gone?
- Lessons learned
- Key enablers
- Ongoing challenges





DISCLAIMER

FINAL VERSION ~~MAY~~ WILL BE DIFFERENT

(WE ARE STILL EVOLVING, AND SO IS SRE)



Tenet Framework: What Do Our SRE's Do?

North Star Framework: SRE Tenets & Sub Tenets

Reliability	Scalability	Operability	Observability
Software shortcomings and technical debt <ul style="list-style-type: none"> Bugfixes for incidents Fault tolerant/defensive design/coding 	Open-Source/Architecture Migration <ul style="list-style-type: none"> Database migration to Open Source based on risk assessment Monitoring toolsets migration towards cost 	Virtual Assistant <ul style="list-style-type: none"> Static and dynamic queries Natural language processing API's for security and system integration 	Customer Focus <ul style="list-style-type: none"> Service reliability drives development velocity using SLI, SLOs, and error budgets XLA's are used to quantify customer experience
Auto Heal <ul style="list-style-type: none"> Automated recovery for service failures Persistence for critical data loss avoidance 	Throughput Tuning <ul style="list-style-type: none"> System performance tuning – latency hotspots Transversal performance tuning 	Robotization <ul style="list-style-type: none"> Toil automated as it is identified Low priority incident reduction Automated Archival 	Auto Detect - Advanced diagnostics <ul style="list-style-type: none"> Log and metric aggregation across service Faster scenarios pattern recognition
Zero-downtime Deployment (Always On) <ul style="list-style-type: none"> Release deployment using "Always On" pattern Microservices and cloud deployment Incremental canary releases 	Scaling - Horizontal & Vertical <ul style="list-style-type: none"> Vertical Scaling – Add RAM/DISK/CPU to Virtual Machine Horizontal Scaling – Add new nodes in cluster 	Tools Engineering <ul style="list-style-type: none"> Productivity toolsets Service recovery toolsets 	Machine learning <ul style="list-style-type: none"> Data science for potential failure prediction
Fault tolerant services <ul style="list-style-type: none"> Circuit breaker - routing to healthy route Chaos Engineering 	Capacity forecasting & Provisioning <ul style="list-style-type: none"> Predictive infra capacity reviews Predictive Application/Service capacity reviews 	SRE Frameworks & Design Patterns <ul style="list-style-type: none"> Standardized SRE supported frameworks used to build systems "Production ready" by design 	Telemetry & Instrumentation <ul style="list-style-type: none"> Exception alerting Capacity and Performance metrics
Recovery Optimization <ul style="list-style-type: none"> Incident resiliency assessment Design remediation and recovery automation 	Automated failover <ul style="list-style-type: none"> Seamless DR Failover for Application/Service Optimize clicks for DR 		Transaction traceability <ul style="list-style-type: none"> Track/map unique identifier across service Transaction and Performance tracking
Automated deployment <ul style="list-style-type: none"> IT Orchestration – Task sequencing & Event scheduling Deployment management, Environment provisioning 			Code Profiling <ul style="list-style-type: none"> Benchmark current performance, identify hotspots and remediate performance

“SRE” is applicable to many different Engineering roles.

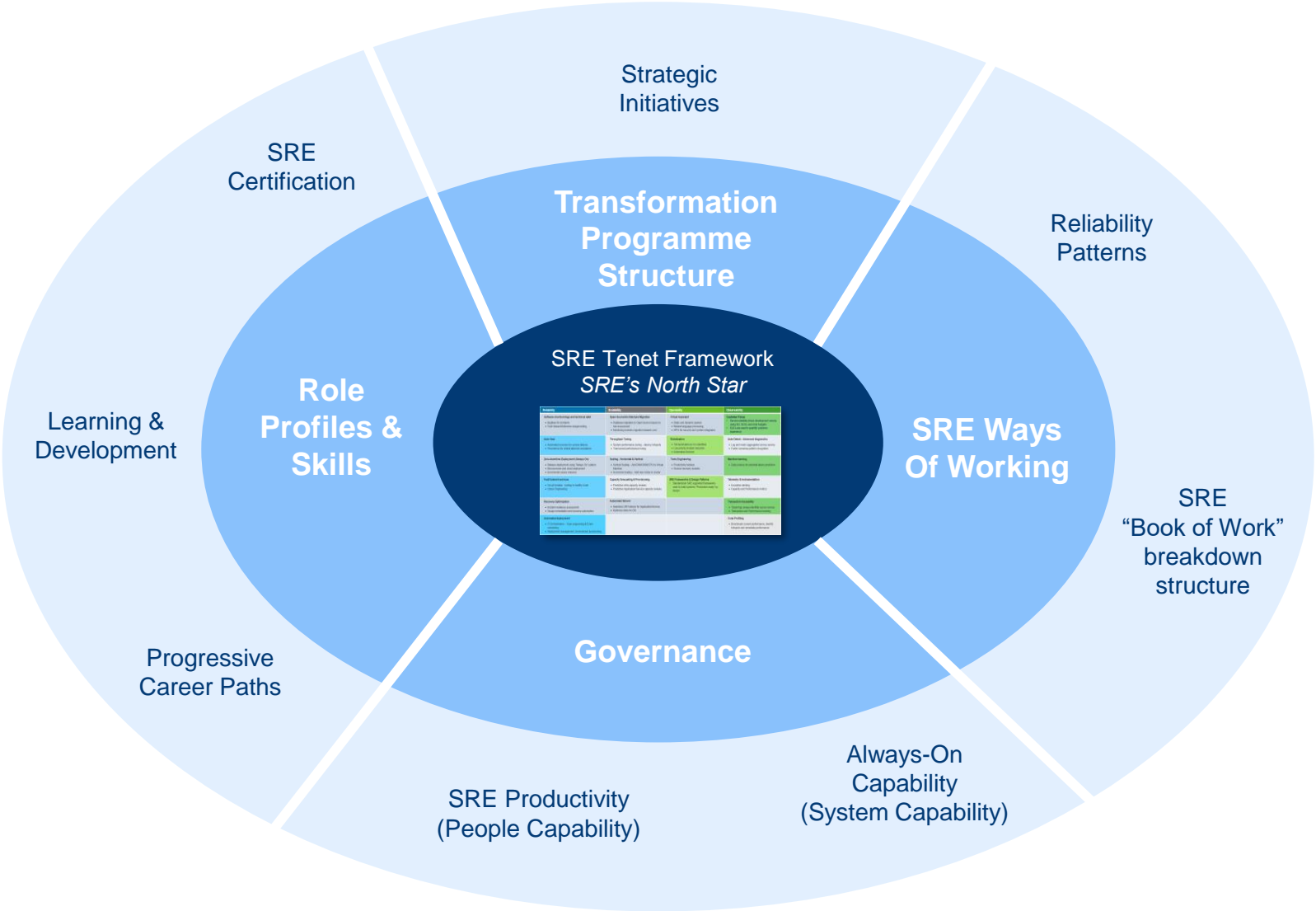
It is important to define early on **what your flavour of SRE does**

Our Tenets were stolen with pride from Google, and **tailored for CCIB's “System” flavour of SRE**, supporting critical business applications



Anchoring Our Transformation

SRE Tenets Form The Core Of Our SRE Operating Model



All aspects of the SRE Operating Model **have lineage back to the SRE Tenets**

Throughout our transformation we have referred to the SRE Tenet framework **to validate our path**

Just as ITIL provides structure for Service Mgmt., the **SRE Tenets** provide structure for our **SRE Operating Model**



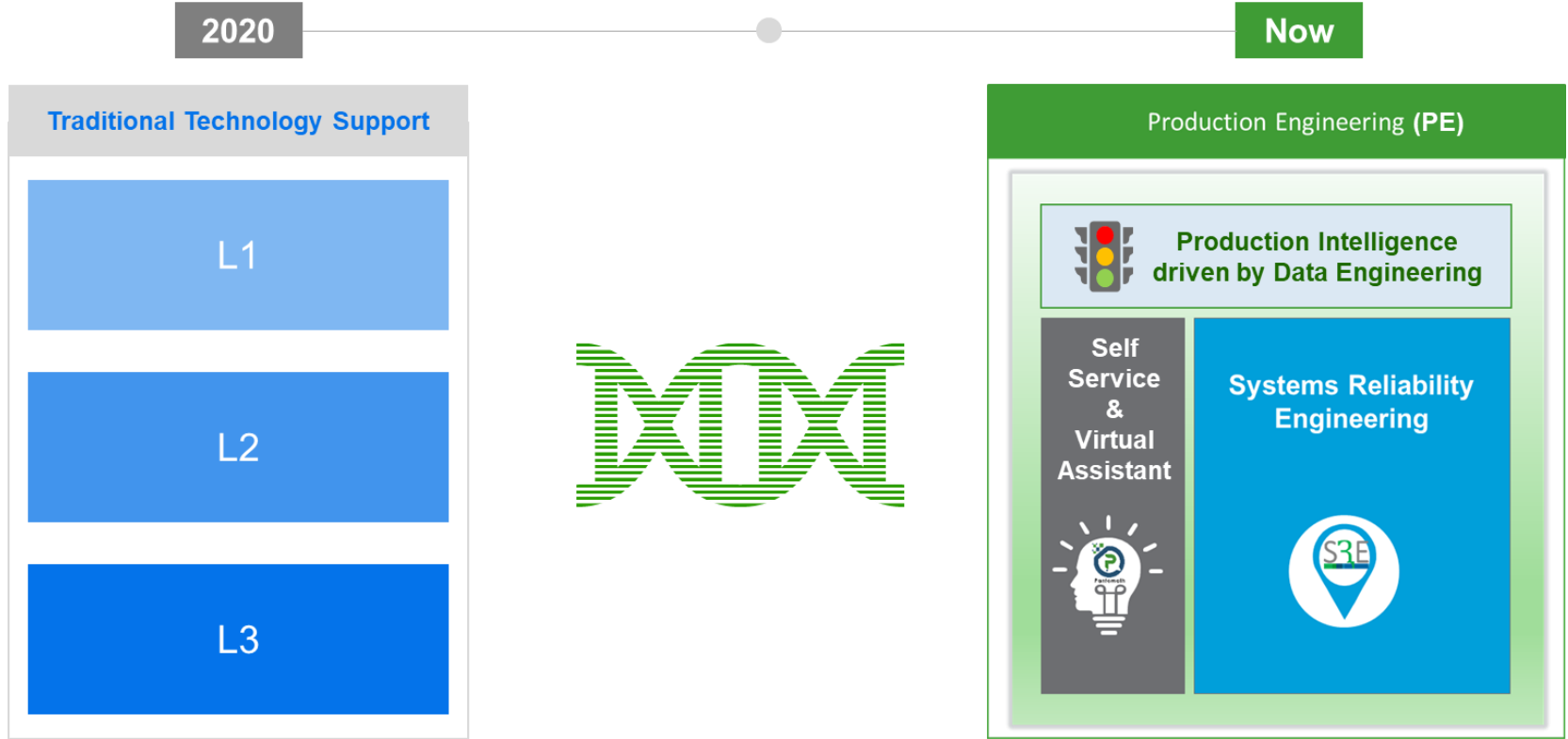
1. People

(Learning & Development)



Our Capability Change From Operations To Engineering

Re-wiring Application Engineering culture from re-active to pro-active



Traditional L1-L3 support functions **will struggle to support “Always-On” digital client journeys**

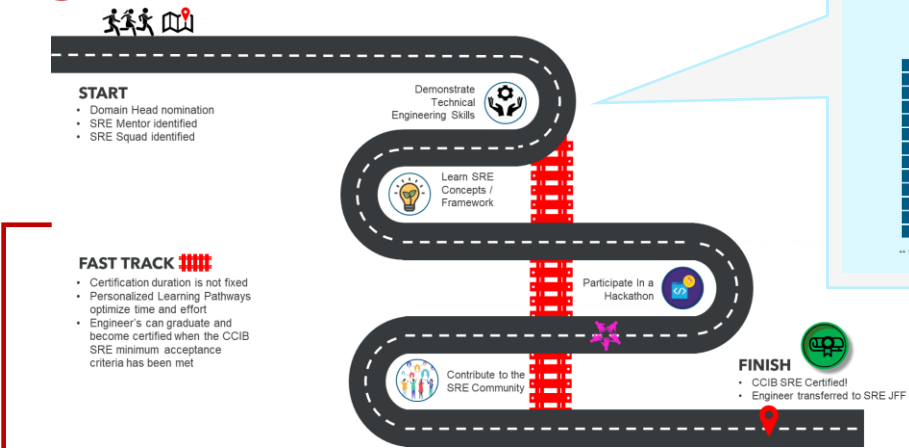
Recalibration was required to support the business demands



Bridging The Capability Gap

SRE Education, Certification & Career Pathways

A SRE Education & Learning Pathway



Essential SRE Discipline (Mandatory To All T&I SRE's)		Specialist SRE Disciplines (Electives, Selected By SRE / Domain)	
Discipline	ECP Skill Level	Specialist Elective	ECP Skill Level
Programming**	Intermediate	Specialist Elective 1	Advanced
Programming Frameworks	Intermediate	Specialist Elective 2	Advanced
Scripting	Intermediate	Specialist Elective 3	Expert
Operating System	Intermediate	Specialist Elective 4	Expert
APIs	Intermediate		
Database**	Intermediate		
Cloud	Foundation		
Compute	Foundation		
Web	Intermediate		
Networking	Foundation		
CI/CD	Intermediate		
Monitoring	Intermediate		
Integration	Intermediate		

1 Advanced Skill = 5 Credits
1 Expert Skill = 10 Credits

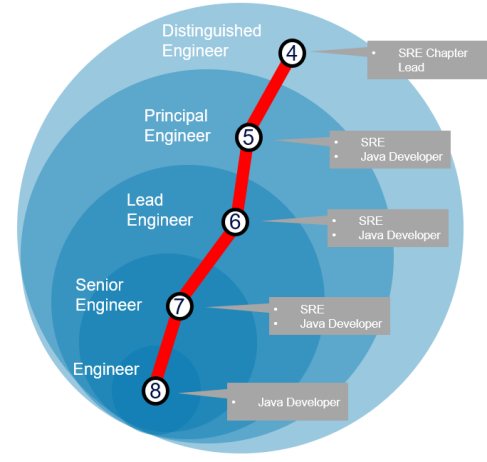
** Those pursuing the SRE 'Lite' tier are only required to demonstrate this discipline to Foundation level

A Make sure SRE's have the skills to be successful in an SRE role

B Create a transparent benchmark for what it means to be an SRE, which incentivises continuous learning

C CCIB SRE Career Pathway aligned to Engineering Career Pathway (ECP) Framework

- █ Progressive Engineering Track
- X Corporate Grade (Indicative Only)
- Examples Of Functional Roles



C Curate an SRE career path that has longevity, without requiring line management



A Bit More On Learning Pathways..



Skills Catalogue

Skill Levels			
Foundation	Intermediate	Advanced	Expert
<p>Proficiencies:</p> <ul style="list-style-type: none"> Understand concepts: <ul style="list-style-type: none"> Control flow language syntax Basic data structures, collections Exceptions and error handling Generics Be able to: <ul style="list-style-type: none"> Read and understand code Write basic algorithms in Java Core libraries (lang, utils) Use unit-testing tool JUnit Setup development environment, specific to the bank Understand basic build environment (eg pom) Understand concepts: <ul style="list-style-type: none"> Skill levels Proficiencies Be able to: <p>Learning pathway:</p> <p>Course: Java Foundation</p>	<p>Proficiencies:</p> <ul style="list-style-type: none"> Understand concepts: <ul style="list-style-type: none"> Java Virtual Machine Memory allocation and garbage collection Concurrency Object-Oriented Programming: Objects, Encapsulation, Interfaces, Classes, Polymorphism Application configuration GoF patterns Be able to: <ul style="list-style-type: none"> Use APIs: Threads, JDBC, HTTP clients, JMS, Servlets Externalize configuration e.g. connection strings and secrets Measure test code coverage Refactor long methods and code duplication Debug applications on local machine Deploy applications on Java application server or as standalone process <p>Learning pathway:</p> <p>Course: Java Intermediate Book: Effective Java, Josh Bloch</p>	<p>Proficiencies:</p> <ul style="list-style-type: none"> Understand concepts: <ul style="list-style-type: none"> Java security and public key infrastructure Java release cycle Application architecture e.g. Layered Architecture Design principles: coupling and cohesion Lambda Expressions Metrics and tracing Be able to: <ul style="list-style-type: none"> Use APIs: Streams, Dependency Injection (CDI), JPA, JMS, NIO Design modular systems Measure code quality metrics in SonarQube Refactor large classes Debug and troubleshoot applications on remote server <p>Learning pathway:</p> <p>Course: Java advanced</p>	<p>Proficiencies:</p> <ul style="list-style-type: none"> Understand concepts: <ul style="list-style-type: none"> Java module system Systems Architecture: Microservices, Enterprise Integration Patterns Be able to: <ul style="list-style-type: none"> Profile and optimize application performance Refactor large legacy systems <p>Learning pathway:</p> <p>Course: Java expert</p>

Within the Disciplines, skill proficiencies are defined at 1-4 skill levels (not all skills have all 4 levels)

Skills pathways..

- Empower self learning
- Create transparency what is expected



2. Process

(SRE “Way Of Working”)



People and L&D 

So what?





We still had a **challenge**....

- SRE was a new role to the Bank
- With no internal reference points
- We're training 500 people to be "SRE"
- **But there's no established SRE "way of working"**



Two key aspects of our SRE Way Of Working....

1. How we **measure** Reliability

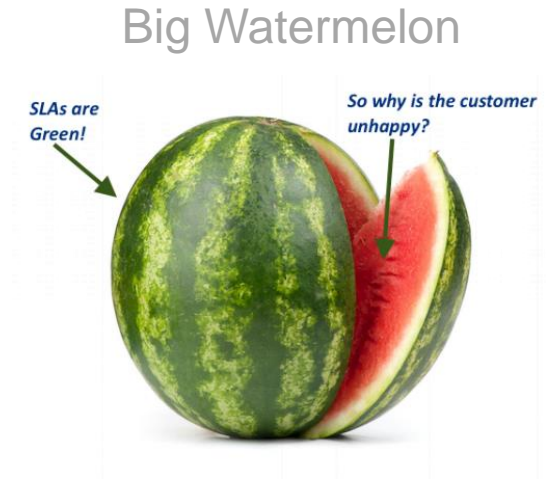
2. How we **elevate** Reliability



Step Back: How is Reliability Measured

Before SRE:

“Technical Availability” or SLAs were leading indicators of reliability

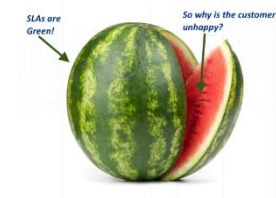


With SRE:

SLI / SLOs measure outcomes customers care about

A significant step forward in measuring reliability through a customer lens

Small Watermelon



Problem: Because How Do We Know...

How a customer **really feels**..?

Or if the **outcomes** a customer wants **change**..?

Or if customer expect **different service levels**?

Observability is the practice of understanding the “**internal state**”

SLI / SLOs are traditionally internal **proxies** for customer sentiment



Opportunity: Qualitative Data

Valuable sources from which Reliability can be inferred

Net Promoter Scores

Customer Satisfaction Surveys

Customer Queries

Customer Effort Surveys

Social Media Listening

Customer Complaints



“Your Users, Not Your Monitoring, Decide Your Reliability”

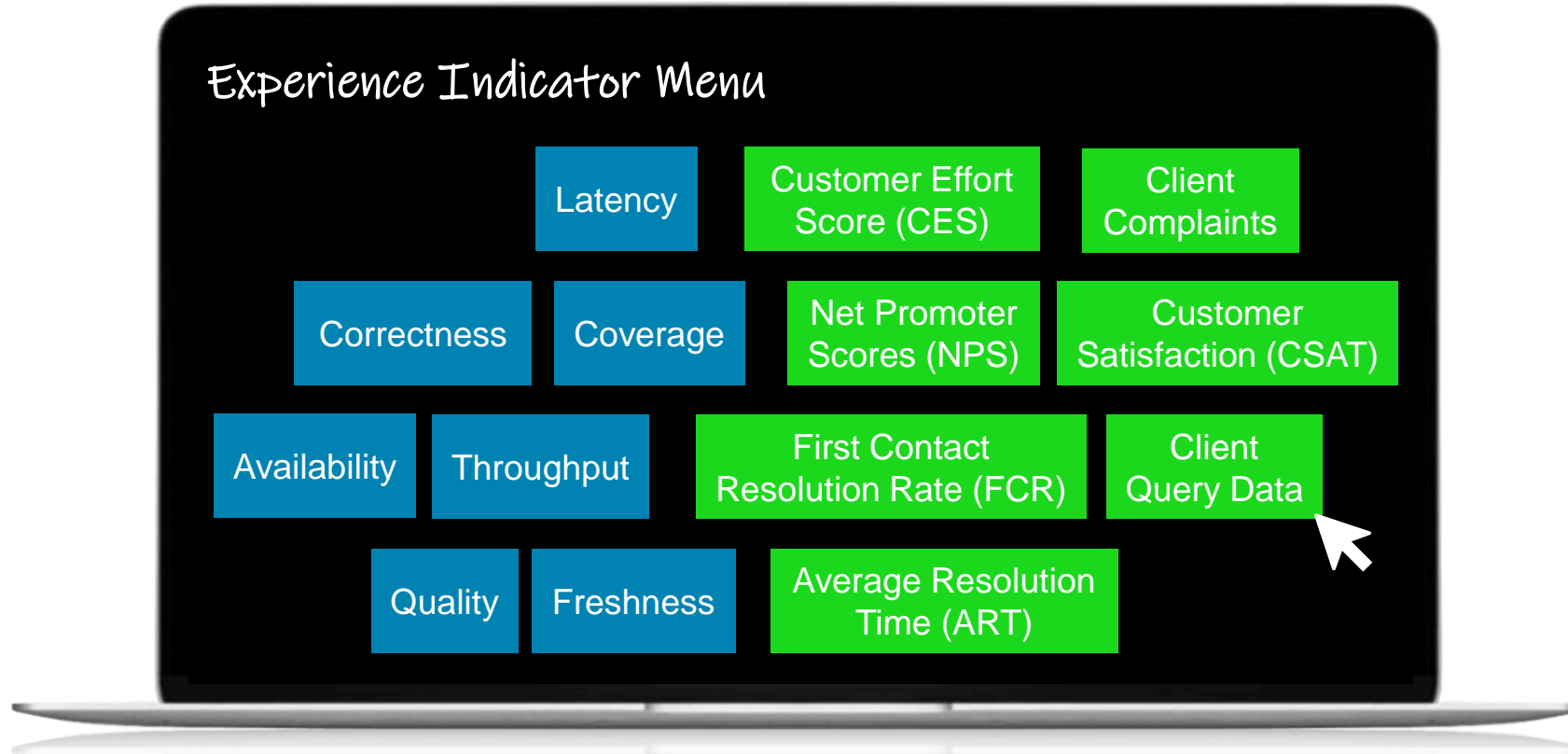
Dave Rensin,
Site Reliability Workbook
Chapter 19: Reaching Beyond Your Walls

If **customers define Reliability**.. we should **expand our evaluation of Reliability** to include **data received direct from the customer**



Solution: Customer Experience Agreements

Extend SLI SLO concept to qualitative data to provide a 360° view



 Technical quantitative service level indicators (Google's SLI Menu)

 Quantitative interpretations of **qualitative** data points

Qualitative metrics enable us to validate our **internal** technical understanding **vs. the customer's perspective**

Our **Product / Business Owners** select the most appropriate "indicators" from a menu of quantitative and qualitative measures



Influencing Investment & ROI

Fast “Immediate” Payment Ecosystem



Recommendation #1 – Corporate Channel Authorizer Workflows

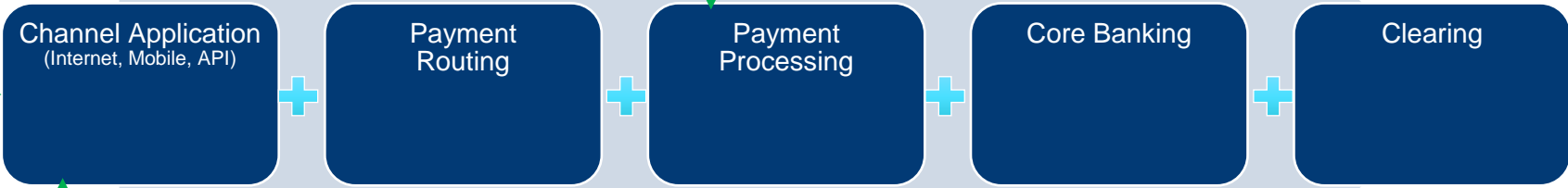
Description: Payment authorizer workflow errors are not described in business English, impacting payment release
Business Impact: Client experience impacted and missed cutoffs due to failure to approve payment on time
of occurrences: 30 P5 / month
Solution: Enhance UI to provide clients with the root cause of each authorizer error
Priority: High



Recommendation #3 – Payment Routing

Description: Single payments are being routed via the Bulk payment channel
Business Impact: Single payment latency degraded while single payments queue behind large bulk files
Solution: Re-route single payments via single payment channel
Priority: High

Payment initiated



Transfer of funds complete



Recommendation #2 – Corporate Channel Bulk Uploads

Description: Bulk Upload error not relayed back to client
Business Impact: Client experience impacted and delay in payment processing while client contacts bank for details
of occurrences: 30 P5 / month
Solution: Enhance UI to provide clients with failure reason
Priority: High



Recommendation #4 – Core Banking Throughput

Description: Delays to payment processing occur during high volume due to limited queue prioritization and throughput capacity
Business Impact: High payment latency
of occurrences: 2.8K payments daily
Solution: Re-validate payment queue prioritization, increase MIPs for Core Banking application
Priority: High

Product ecosystems can be opaque (it's bad but why?) leading to a poor ROI in customer experience

Ecosystems also often have multiple owners with their own priorities

Blending quantitative and qualitative data and visualizing the opportunities E2E has helped us break down silos and influence investment



Challenges Encountered Measuring Reliability

- Measuring using Functional Traces is complex – edge cases, “watermelon”
- Aggregation – data can be spread over various organization silos, proliferation of operational tools
- Granularity Trade-offs – how low level do you go?
- Suitability of Qualitative data (for quantitative evaluation)
- Scaling – Product driven (Fast Payments) vs. Metric driven (Latency)

The key to bursting the “watermelon” is to ensure the Product or Business Owner is involved shaping the trade-offs



Wrapping Up: "Measuring Reliability" Lifecycle

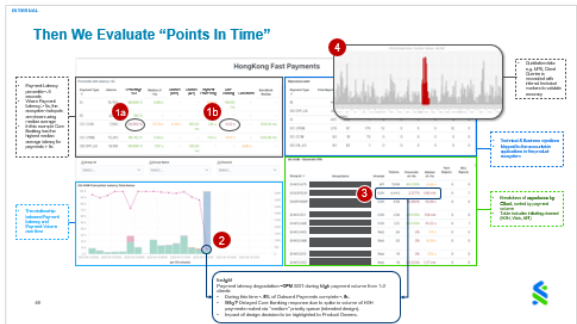
1

Customer Experience is evaluated end to end for a product's ecosystem using a blend of Quantitative & Qualitative indicators



2

Real-time quantitative Customer Experience data is embedded in Technology & Operations



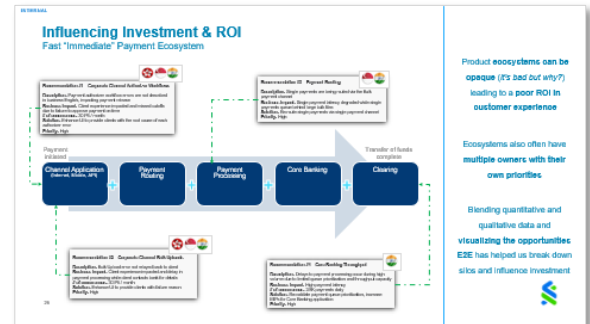
3

Customer Experience data is then aggregated to create "Point in Time" Insights



4

Leading to targeted opportunities to influence investment decisions



We have **extended the SLI / SLO** concept to include **qualitative data**, and measure client experience end to end for key product ecosystems.

Client experience is elevated through **targeted Product Owner recommendations** to maximise ROI.



Two key aspects of our SRE Way Of Working....

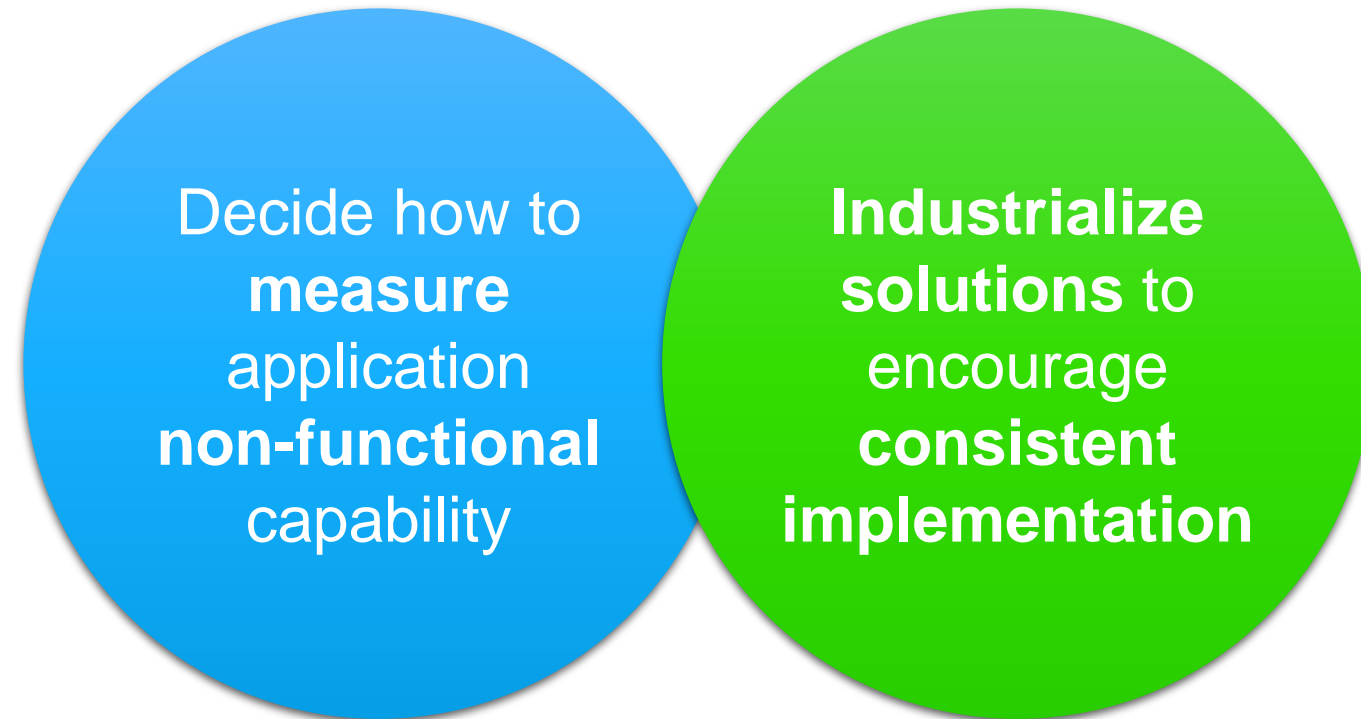
1 .How we **measure** Reliability



2. How we **elevate** Reliability



2. How we elevate Reliability?



Recap: Tenet Framework: SRE Focus Areas

North Star Framework: SRE Tenets & Sub Tenets

Reliability	Scalability	Operability	Observability
Software shortcomings and technical debt <ul style="list-style-type: none"> Bugfixes for incidents Fault tolerant/defensive design/coding 	Open-Source/Architecture Migration <ul style="list-style-type: none"> Database migration to Open Source based on risk assessment Monitoring toolsets migration towards cost 	Virtual Assistant <ul style="list-style-type: none"> Static and dynamic queries Natural language processing API's for security and system integration 	Customer Focus <ul style="list-style-type: none"> Service reliability drives development velocity using SLI, SLOs, and error budgets XLA's are used to quantify customer experience
Auto Heal <ul style="list-style-type: none"> Automated recovery for service failures Persistence for critical data loss avoidance 	Throughput Tuning <ul style="list-style-type: none"> System performance tuning – latency hotspots Transversal performance tuning 	Robotization <ul style="list-style-type: none"> Toil automated as it is identified Low priority incident reduction Automated Archival 	Auto Detect - Advanced diagnostics <ul style="list-style-type: none"> Log and metric aggregation across service Faster scenarios pattern recognition
Zero-downtime Deployment (Always On) <ul style="list-style-type: none"> Release deployment using “Always On” pattern Microservices and cloud deployment Incremental canary releases 	Scaling - Horizontal & Vertical <ul style="list-style-type: none"> Vertical Scaling – Add RAM/DISK/CPU to Virtual Machine Horizontal Scaling – Add new nodes in cluster 	Tools Engineering <ul style="list-style-type: none"> Productivity toolsets Service recovery toolsets 	Machine learning <ul style="list-style-type: none"> Data science for potential failure prediction
Fault tolerant services <ul style="list-style-type: none"> Circuit breaker - routing to healthy route Chaos Engineering 	Capacity forecasting & Provisioning <ul style="list-style-type: none"> Predictive infra capacity reviews Predictive Application/Service capacity reviews 	SRE Frameworks & Design Patterns <ul style="list-style-type: none"> Standardized SRE supported frameworks used to build systems “Production ready” by design 	Telemetry & Instrumentation <ul style="list-style-type: none"> Exception alerting Capacity and Performance metrics
Recovery Optimization <ul style="list-style-type: none"> Incident resiliency assessment Design remediation and recovery automation 	Automated failover <ul style="list-style-type: none"> Seamless DR Failover for Application/Service Optimize clicks for DR 		Transaction traceability <ul style="list-style-type: none"> Track/map unique identifier across service Transaction and Performance tracking
Automated deployment <ul style="list-style-type: none"> IT Orchestration – Task sequencing & Event scheduling Deployment management, Environment provisioning 			Code Profiling <ul style="list-style-type: none"> Benchmark current performance, identify hotspots and remediate performance

The Tenet Framework contains **key SRE focus areas** and by extension, the **non functional capability we expect to see in our critical business applications**



Decide how to measure application capability

Non-functional capabilities described with lineage back to SRE Tenets

Lineage back to "North Star" framework


Tenet Framework: What Do Our SRE's Do?
North Star Framework: SRE Tenets & Sub Tenets

Category	Sub-Tenet	Objective	Guidelines
Reliability	Always On Deployment Architecture	Open Source/Offshore/Third Party	High Availability
Availability	High Availability	High Availability	High Availability
Performance	Performance	Performance	Performance
Security	Security	Security	Security
Compliance	Compliance	Compliance	Compliance

"SRE" is applicable to many different Engineering roles.

It is important to define early on what your flavour of SRE does

Our Tenets were stolen with pride from Google, and tailored for CCIB's "System" flavour of SRE, supporting critical business applications



Specifications

Always On Deployment Architecture

Element ID	Purpose/Objective	Implementation Guidelines & Design Pattern
AODA-ZDD-000	Intraday Change deployment without downtime is practiced for all application components supporting a critical service.	
Element ID	Capability Element	Implementation Guidelines & Design Pattern
AODA-ZDD-001	Web Tier Always On	<p>Objective : No downtime is required to deploy changes for services within agreed operating hours</p> <p>Methods/Tools : Rolling Deployment/Blue-Green deployment/Canary Deployment Deployment Stamps Pattern Zero Downtime</p> <p>Maturity Measurement Data Source : Application Availability Report</p>
AODA-ZDD-002	Application Tier Always On	<p>Objective : No downtime is required to deploy changes to database objects within agreed operating hours</p> <p>Methods/Tools : Redundancy/Clustering Database High Availability</p> <p>Maturity Measurement Data Source : Database Availability Report</p>
AODA-ZDD-003	Database Tier Always On	<p>Objective : No downtime is required for hygiene activities e.g., Patches, Reboots</p> <p>Methods/Tools : Rolling Deployment/Blue-Green deployment/Canary Deployment Zero Downtime</p> <p>Maturity Measurement Data Source : Application Availability Report</p>
AODA-ZDD-004	Hygiene (OS, DB Patches, Reboots) Always On	<p>Objective : All critical online functions are available during batch runtimes</p> <p>Methods/Tools : Service Load Balancing Queue-Based Load Leveling Sharding Pattern</p> <p>Maturity Measurement Data Source : Application Availability Report</p>
AODA-ZDD-005	Batch does not disrupt Online	<p>Objective : Cloud Node Group / Cluster upgrade refresh should be automated and deployed without any downtime for the production accounts</p> <p>Methods/Tools : Redundancy/Clustering Container Cluster Upgrade</p> <p>Maturity Measurement Data Source : Application Availability Report</p>
AODA-ZDD-007	Canary Testing implemented	<p>Incremental canary testing possible for 100% of application functions supporting a critical service journey</p> <p>Deprecated in Version 2. Merged with AODA-ZDD-001 implementation guide.</p>
AODA-ZDD-008	Cloud Node Group / Cluster upgrade or refreshes do not incur application downtime	

Agree a methodology to evaluate non functional capability.

The framework drives daily SRE activities, and provides structure for less experienced Engineers

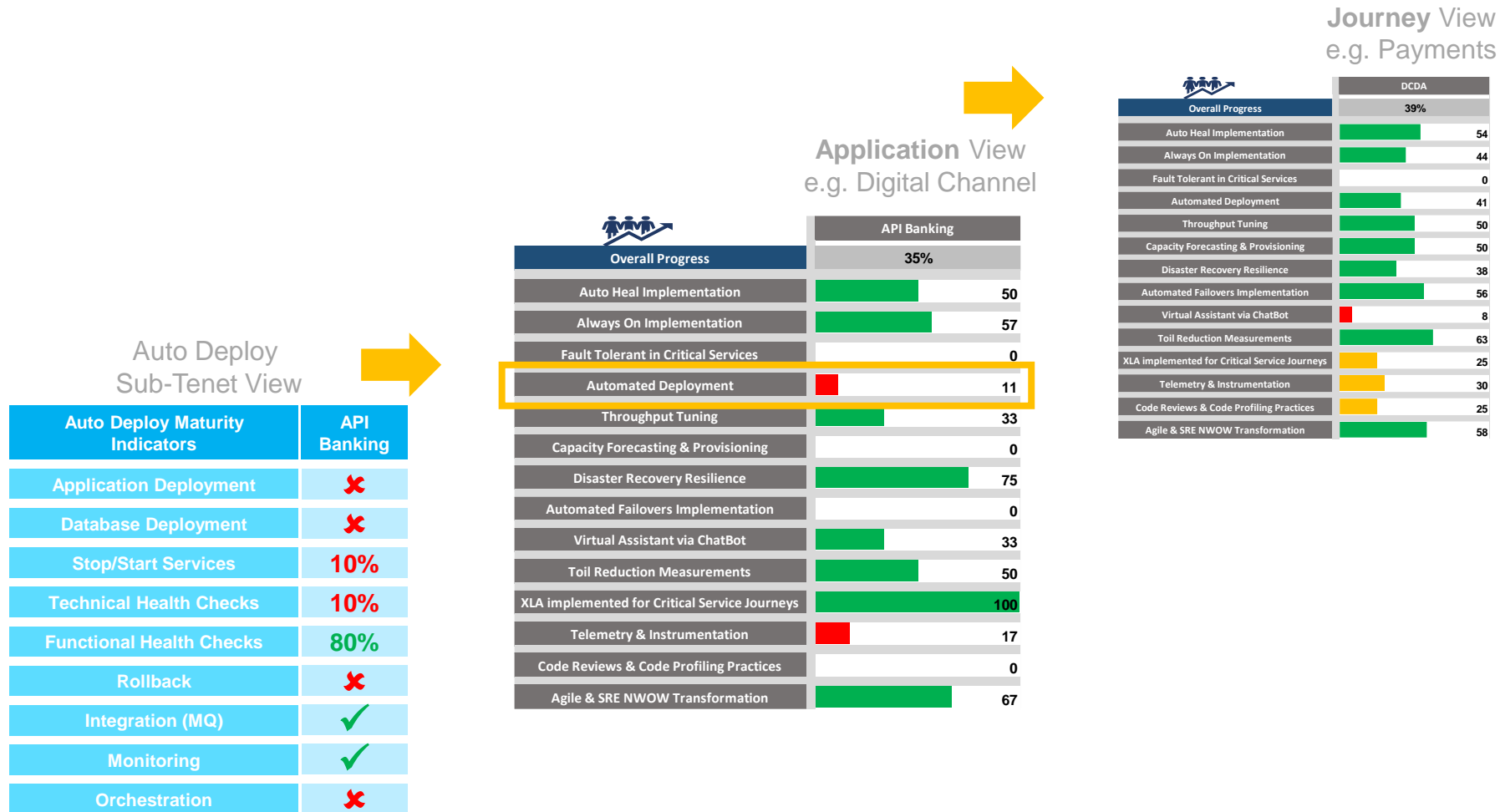
Our non-functional framework is similar in intent to R9Y

<https://map.r9y.dev/beck/map.html>



Then Visualize Capability E2E

A service is only as strong as it's weakest link



When evaluating, where possible **remove subjectivity** and **focus on measuring outcomes**

e.g. we are experimenting with a combination of technical markers and ITSM data

Crowd Sourced Feedback:

How do you evaluate an applications non-functional capability?



Reliability Patterns: Industrializing solutions

Non-functional capabilities (each with lineage back to SRE Tenets)

Element ID	Capability Element	Implementation Guidelines & Design Pattern
RECT-TMI-000	Observability Telemetry & Instrumentation	Implement observability solutions to aggregate logs, infrastructure metrics, application performance and transaction data to an integrated platform to be able build end-to-end service journey observability dashboard and create alerts for failures that have imminent impact to critical service journeys
RECT-TMI-001	Monitoring Baseline Standard Defined & Implemented	<p>Objective: All critical service journeys have their monitoring standards defined and implemented</p> <p>Guide/Tool: Tableau Installation Guide for PIC - Production Engineering - TB_DCDs_CC & CE - Confluence (standardschartered.com)</p> <p>ITRS Genesys Tool</p> <p>Prometheus - Monitoring system & time series database</p> <p>Maturity Measurement Data Source: SkyNet</p>
RECT-TMI-002	Logs are aggregated for on prem / cloud instances into a single observability platform	<p>100% coverage</p> <p>Guide/Tool: SkyNet Architecture - Blueprint - Production Engineering - TB_DCDs_CC & CE - Confluence (standardschartered.com)</p> <p>Maturity Measurement Data Source: SkyNet</p>
RECT-TMI-003	Infra metrics are aggregated for on prem / cloud instances into a single observability platform	<p>100% coverage</p> <p>Guide/Tool: Teneigh/TSDM</p> <p>Maturity Measurement Data Source: SkyNet</p>
RECT-TMI-004	Application performance data is aggregated for on prem / cloud instances into a single observability platform	<p>100% of runtime (e.g. JVM CLR, Google V8, Containers) monitored by a performance management tool</p> <p>Guide/Tool: Elastic APM setup for JIRA & Jiraconet - Production Engineering - TB_DCDs_CC & CE - Confluence (standardschartered.com)</p> <p>10 BEST APM Tools (Application Performance Monitoring Tools in 2023) (softwareinsights.com)</p> <p>Maturity Measurement Data Source: SkyNet</p>
RECT-TAF-001	Critical Service Journey's Mapped, Traceable, E2E and aggregated into single observability platform and integrated with intelligent tool (e.g. Siren, Pager Duty)	<p>100% of critical service journey's documented end to end</p> <p>Guide/Tool: Siren - Production Engineering - TB_DCDs_CC & CE - Confluence (standardschartered.com)</p> <p>Integrations: Integrate With PagerDuty / PaperDuty</p> <p>Maturity Measurement Data Source: SkyNet</p>

Reliability Pattern e.g. logging standard



Sample log4j config checked into Repository

“If everyone moves forward together then success takes care of itself”

Henry Ford

Our SRE’s leverage and contribute to the in-house developer reference framework

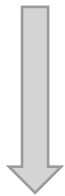
Reward innovation through collaboration and re-use (in addition to rewarding creation)



Wrapping Up: SRE Way Of Working Lifecycle

Integrated approach to empower SREs to decide the WHAT and HOW

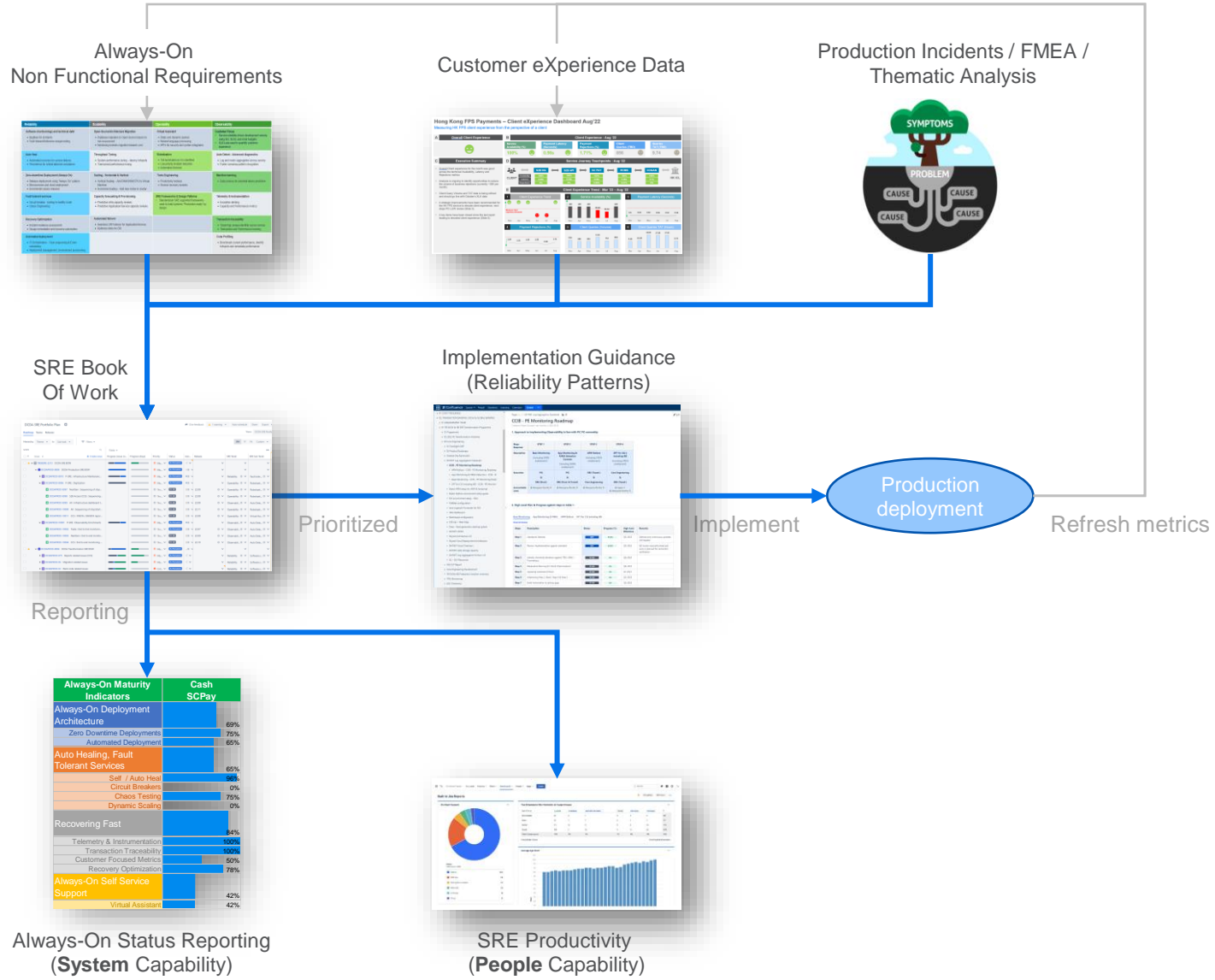
Inputs
to SRE Backlog



Prioritization
& Implementation



Automated
Governance



3. Tools

- Observability
- Automated incident response
- Virtual response



People & Culture 

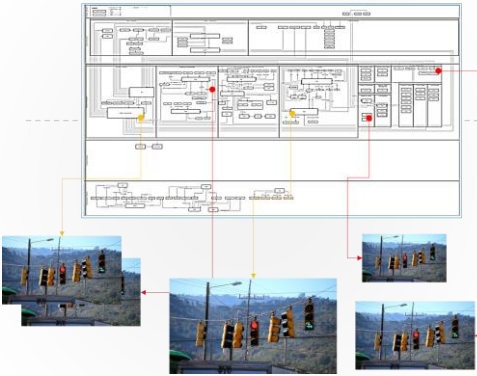
Process 

Some tools?

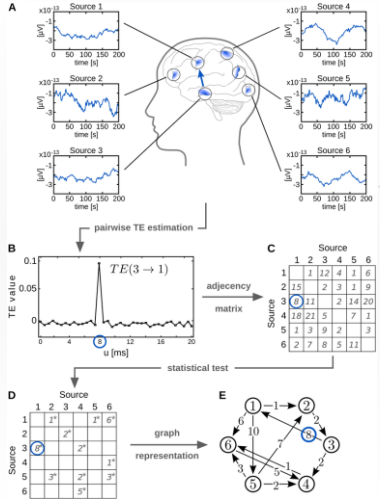


Traditional Monitoring & Signal Processing

Traditional Monitoring



Signal Overload



Failure To Correlate



Traditional monitoring is best suited to **non digital workflows driven by singular uncorrelated signal events.**

This paradigm is **obsolete** in the Digital economy

Signal processing is a complex task **which requires assimilation and inference**



Challenge: Inference & Response



Eyes on Glass



Event / Alert spam



Slow to determine cause & impact

 ~450k pm



Customer(s) often inform of the problem

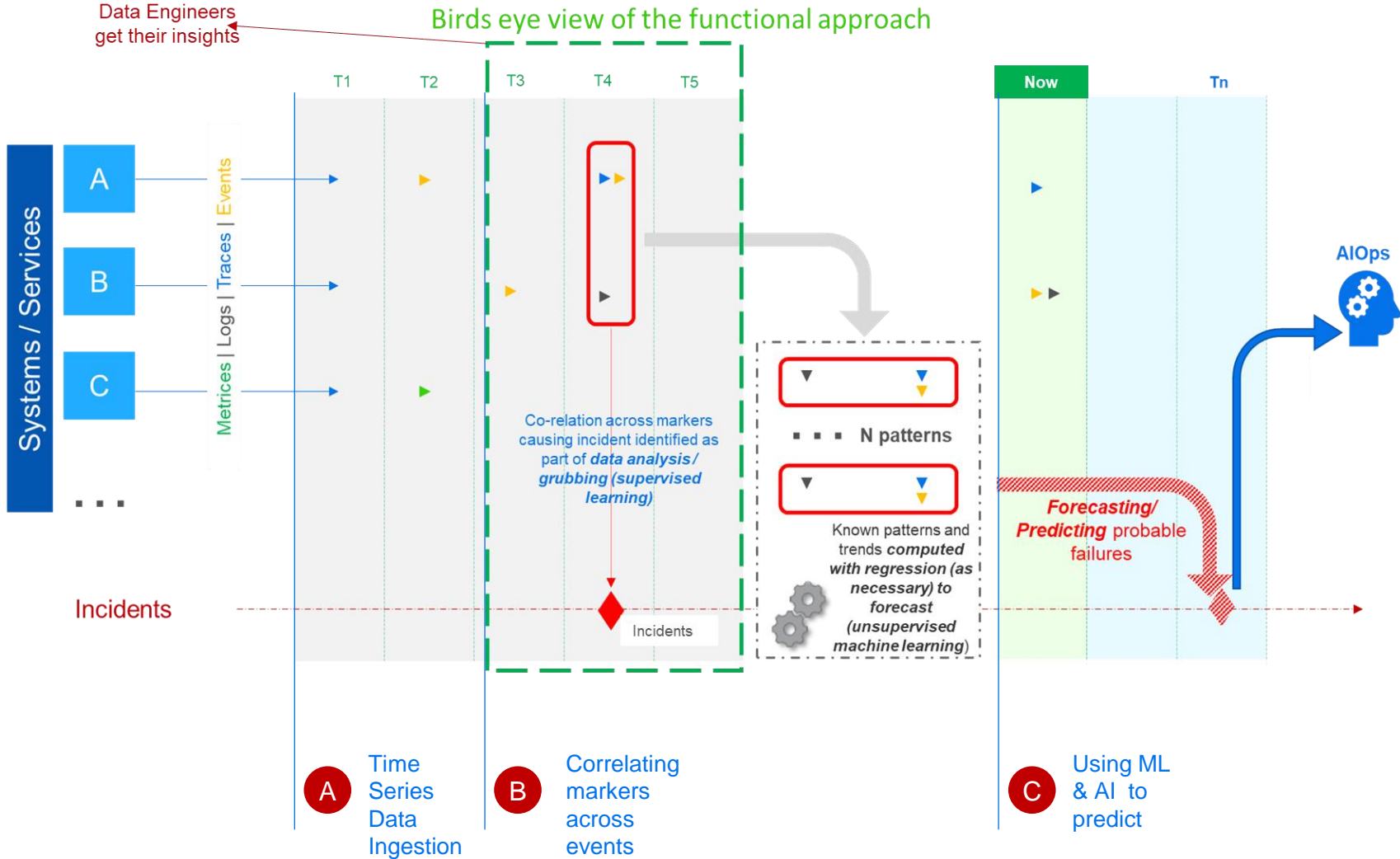


Risk of regulatory breach

Summarized, the impact of legacy **traditional approach to monitoring** was forcing our Engineers to operate in a **reactive mode**, burdened by manual triage and response



Observing Digital Customer Journey's E2E



We staged the implementation in 3 phases to allow for bedding down of the observability platform and making it operable for data engineers



Time Series & Dynamic Tuning To Reduce Noise



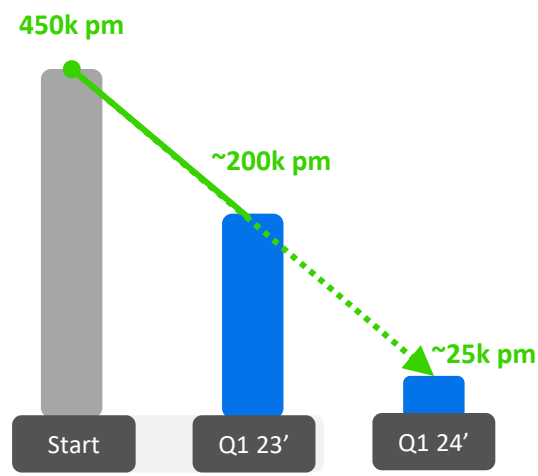
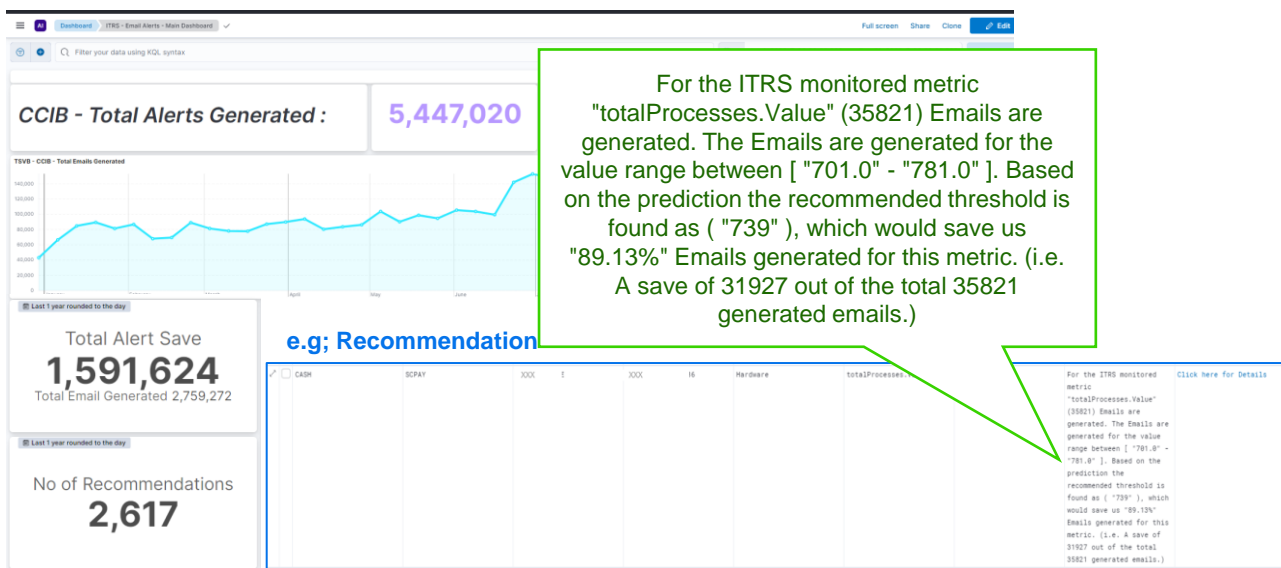
~5.5 million alerts generated from our applications in 2021



Genuine warnings obscured by large number of false / informational alerts



In house ML solution designed to identify common alert patterns by utilizing historical alert records



Ingesting data in Time Series **A** and using ML **C**, allowed us to form K-means cluster to prescribe ranges

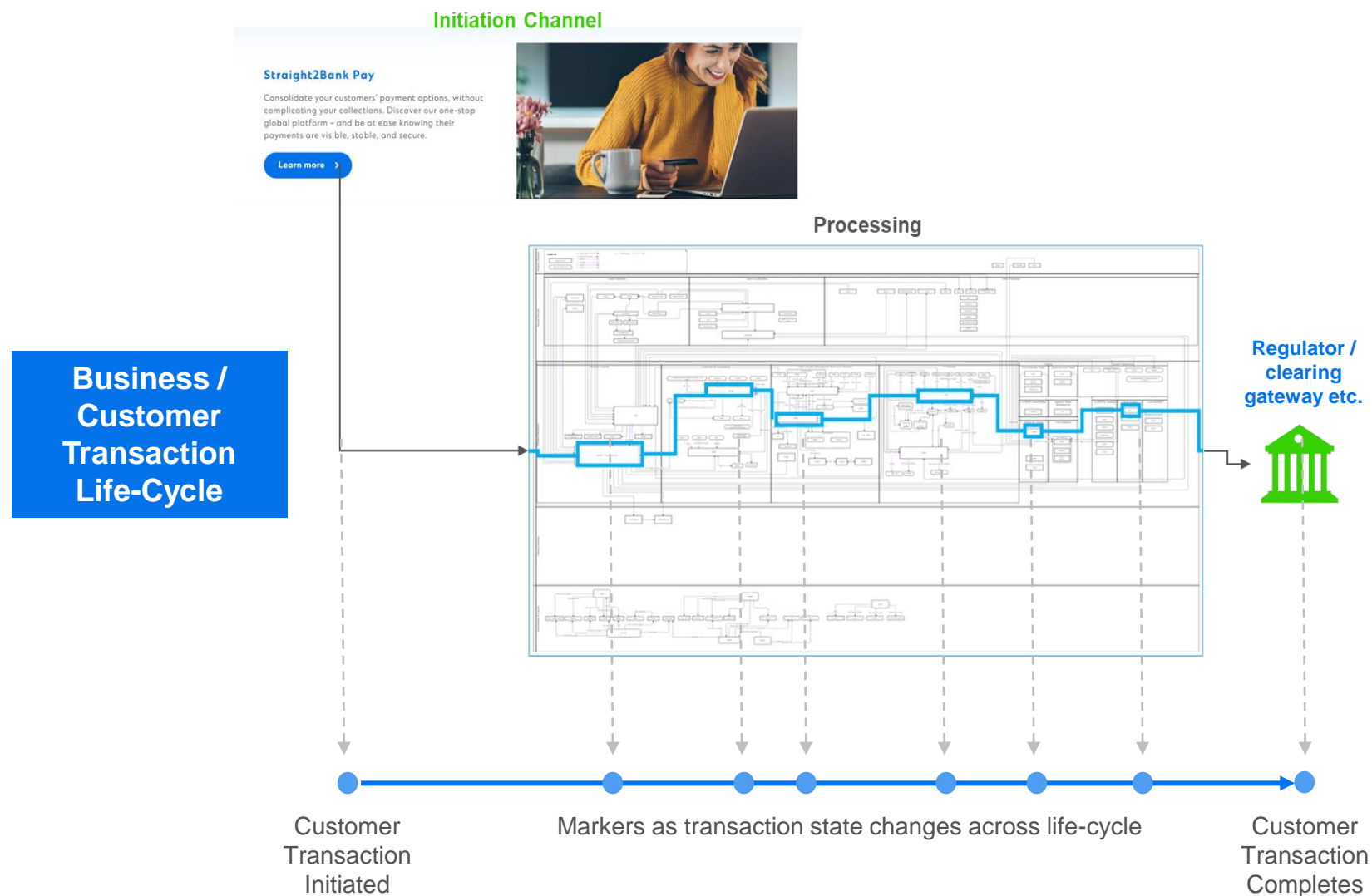
This was then implemented for 30+ crown jewel applications

50+% reduction in false alerts.



Primer: Distributed Functional Tracing (DFT)

Tracing Customer events end to end



While time series brought production data in correlated context of time buckets

We still had to solve for **distributed functional tracing (DFT)** across journeys **interweaved between old and new technology** as

- APM traces were not available E2E
- Instrumentation was non-standard



Solution: DFT Solved With Sidecar Implementation

- 1. Standardized Unique End-to-end Transaction Reference (UETR). UETR is a Swift standard.
- 2. Incorporated instrumentation as a side-car pattern across all key systems

{JSON}

SNO	Variable	Format
1	DateTime	DD-Mon-YYYY HH24:MI:SS.MPF
2	Log_TimeZone	±TZ:TZM
3	Application_Name	Alphanumeric
4	Request_Identifier	Alphanumeric
5	Message_Identifier	Alphanumeric
6	Global_Identifier	Alphanumeric
7	Logging_Event	String
8	Country_Code	XX
9	Service_Name	Alphanumeric
10	Service_Description	Alphanumeric
11	Event_Name	Alphanumeric
12	Event_Header	Alphanumeric
13	Event_Detail	Alphanumeric
14	Event_Status_Code	Number
15	Processing_Status	Alphanumeric
16	Request_Status	Alphanumeric
17	Exception_Code	Alphanumeric
18	Exception_Description	Alphanumeric
19	Mapping_Identifier	Alphanumeric
20	Message_Application	Alphanumeric
21	Audit_Info	Alphanumeric
22	Custom_Property	Key value pair

```

JSON Formatted Log
1  {
2  "Request_Identifier" : "9f2c56a4-3dc3-42ca-8b15-917f23158142",
3  "Global_Identifier" : "9f2c56a4-3dc3-42ca-8b15-917f23158142",
4  "Country_Code" : "SG",
5  "Service_Name" : "Accounting",
6  "Service_Description" : "",
7  "Datetime" : "2020-05-29 17:35:10.139",
8  "Processing_Status" : "CRED",
9  "Event_Name" : "STLS",
10 "Event_Header" : "TMC/SET999",
11 "Event_Detail" : "Transaction posting - Success",
12 "Event_Status_Code" : "10000500000999",
13 "Request_Status" : "Success",
14 "Exception_Code" : "",
15 "Exception_Description" : "",
16 "Mapping_Identifier" : "MSCB20200529SCP00040534",
17 "Mapped_Application" : "MEP",
18 "Log_TimeZone" : "+08:00",
19 "Message_Identifier" : "05b001f6-91d5-431d-9878-2d61a08596e2",
20 "Application_Name" : "ScPay - G3",
21 "Logging_Event" : "INFO",
22 "Audit_Info" : "",
23 "Custom_Property" : {
24   "Remittance_Amount" : "1.00",
25   "Payment_Type" : "IBFT",
26   "Sub_Payment_Type" : "IC",
27   "Sys_Ref_Number" : "9f2c56a43dc342ca8b15917f23158142",
28   "Start_Datetime" : "2020-05-29T17:35:10.118",
29   "End_Datetime" : "2020-05-29T17:35:10.120",
30   "Credit_Datetime" : "2020-01-02T17:00:00.000",
31   "Processing_Datetime" : "2020-05-29T17:35:05.568",
32   "Lane_Flow_Priority" : "HIGH"}
33 }

```

Sidecar patterns are extremely useful to bring to life customer transaction journeys when they propagate across new & old tech stacks involving AIX, mid-range, Cobol etc..

This allowed us to bring both technology and business events in time buckets **A** and correlate them **B**

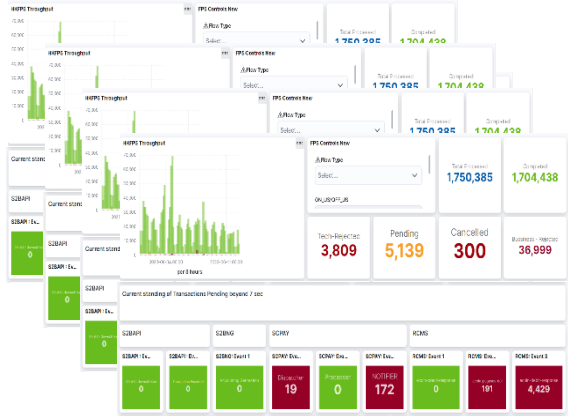
- 3. Real-time ingestion of each transaction state change
- 4. Low level code stitches each transaction state change E2E using UETR



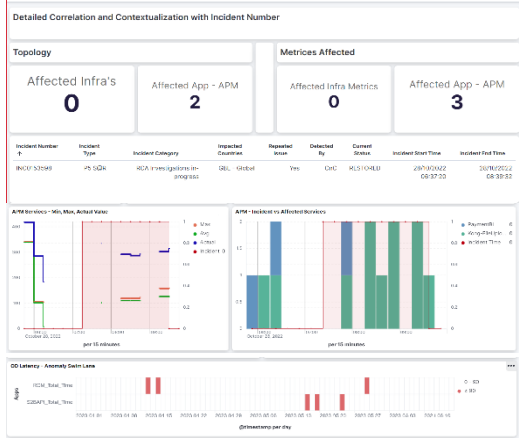
Correlating Business (DFT) & Technology Events



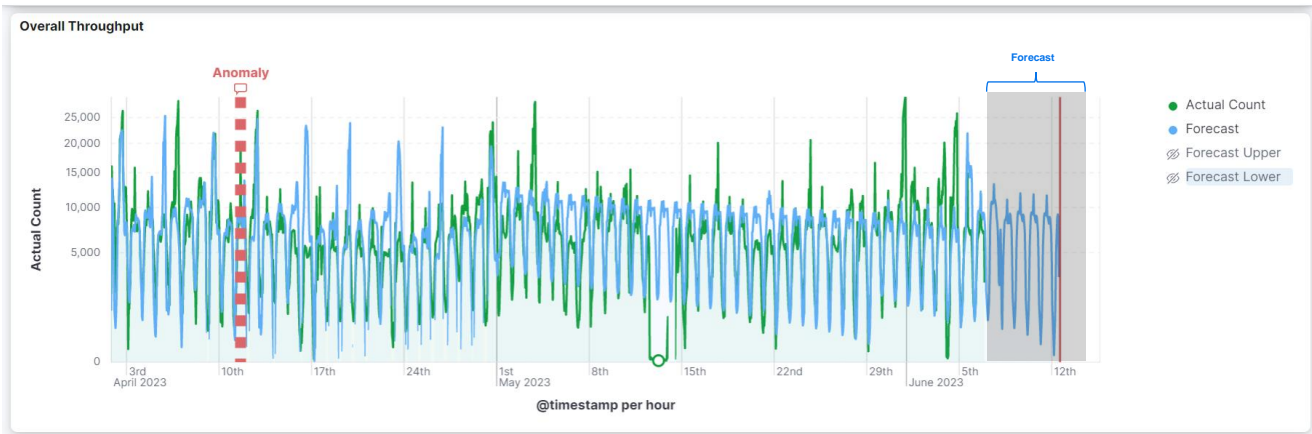
A Data from Multiple Indexes



B Correlation of business transaction challenges to technology events



C Forecasting



We experimented with various ML models like Prophet, LSTM, Isolation forest, Elastic ML. Each had its own pros / cons..

In-house solution developed to sample data and determine most appropriate model to use, culminating in an ensemble of ML models.

This space is changing rapidly with Generative AI being applied to large data sets...



Scale of implementation

Scale

8 Customer journeys and growing..

30+ key platforms

18k microservices

40 nodes

28k documents ingested per second

10B DFT messages stitched and processed in a week

70B documents processed in a week

Stack & Standards



Our Observability platform is now **powering Product Owner investment decisions** for improved customer experience

And our investment in AIOps has improved:

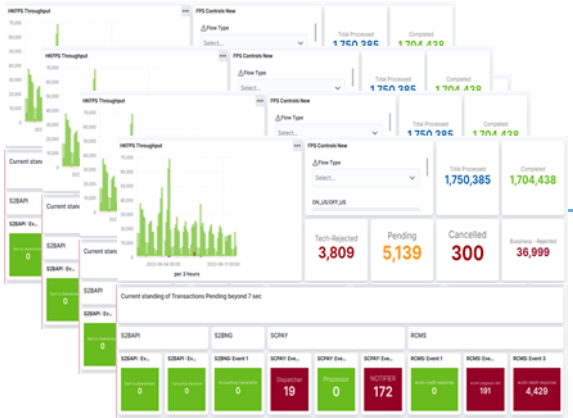
- **MTTD by 75%**
- **MTTR by 25%**



Further Unlocking The Potential Of Production Data

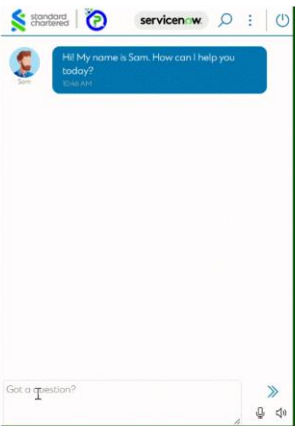
Key enablers which allow our SREs to be SREs

Data from Multiple Indexes



N...

Virtual Assistance



Rapid Triage



Hi User,
 There are 5 Long pending transactions that we did not send the final webhook response (i.e., "rejected" or "completed") back to the client within 1 minute.

To empower self-help
Production data is exposed directly to Business Operations teams using Virtual Assistance and nested APIs

Production data is also leveraged to **facilitate automated intelligent incident response** supported with **correlated root cause insights**



Key Takeaways



standard
chartered

Enabling Our SRE Transformation – Key Takeaways



People

Establish a “**North Star**” guiding SRE framework and create lineage back to it

Invest in L&D and **structured learning pathways**. Then proactively manage the psychology of the change

Neuroplasticity – *“neurons that fire together wire together”*.
Make sure people given the opportunity to apply skills
(*Hackathons, Sandboxes, Workshops*)



Process

Define an operating model & way of working **that empowers SREs to be an SRE**

Extend your evaluation of Reliability to qualitative data received direct from the customer

Start small and **adjust** as momentum builds



Tools

Sidecar patterns can be powerful enablers when used correctly

Invest in Distributed **Functional Tracing**

Observability **enables internal digitization**

Have a centralized Core Engineering **ringfenced from day-to-day operations**



The End

Thank you

