# CYSSC Platform Modernization T661

### What scientific or technological uncertainties did you attempt to overcome? (Maximum 350 words)

￼

We needed a process to move three layer systems — data, business, presentation — from a legacy infrastructure to the cloud.

The Ministry of Children, Community and Social Services (Ministry) mandate includes responsibilities for social assistance, child welfare, and spousal and child support. The Ministry relies on a variety of applications to support its mandate. Of these applications, the three largest are the Social Assistance Management System (SAMS) to deliver social assistance services, the Child Protection Information Network (CPIN) as the consolidated case management system for Children’s Aid Societies that maintains critical information about every child in need of protection, and the Family Case Management System (FCMS) to manage and process payments for spousal and child support. The Common Cluster Platform (CCP) hardware used to house these applications is IBM Power 7/8 infrastructure on which the underpinning technology stack run. The legacy Operating System (OS) is IBM's AIX UNIX variant.

AIX cannot go directly to the cloud. The three applications had to first be ported to Linux, along with the CCP. This is a non-commercial use so we had flexibility with the Linux base. After research and investigation, we selected Red Hat Enterprise Linux (RHEL). We had to design the process in conformity to, *GO-ITS 25.21 Cloud First Principles and Security Requirements*, The Ontario Public Service (OPS) security and compliance standard.

The plan had to include, new ITS GoCloud infrastructure, decommissioning existing Common Cluster Platform (CCP), adopting Cloud technology, performance tuning, and setting up a Disaster Recovery Plan (DRP) inclusive of infrastructure.

The migration — lift and shift — of the AIX hosted applications: SAMS, CPIN, FCMS, mFTIS, as well as operational tools and software such as OEM, Nagios, RTC, Cognos, Control-M, Informatica, and others to the new platform.

Application re-architecture was not in scope. Only changes required to make applications run in the ITS GoCloud platform were part of the process. The first stage we developed was accurate inventory of Ministry requirements and current state of technology. This involved ensuring documentation was accurate and complete. Then we built an alignment on the *target state* infrastructure, including documenting the assumptions and technical constraints. The infrastructure was quite old compared to today’s standards. The state of technology at the beginning of the project had several shortcomings and/or limitations, as well as technological problems and unknown elements.

### What work did you perform in the tax year to overcome the scientific or technological uncertainties described in line 242?

The work we did was the migration. The uncertainty was to sequence. A migration of this scope is like a chess game — if you make a sequence mistake it unfolds as the migration progresses, and you lose the migration and have to go back to mistake as begin again. Our migration went like a house build, foundation first. In this metaphor that meant environment first, environment variables, compilers, development environments, then applications, then data. We wrote our primary migration plan incorporating the various micro-plans for components, like the database transformation and redeployment in RHEL GoCloud context.

APPLICATION MIGRATION

We created a compatibility map with the new infrastructure to integrate the systems. This involved a forensic audit of all the working parts. We created categories — applications replicated already on RHEL, middleware migration by available and needs to be coded, and application mapping for replacements by either coding or a suitable replacement. For each of these component parts we had to map to a duplicate service on GoCloud. Not all the technical parameters and integration points were known.

DATA MIGRATION

The migration was from DB2 to Mongo Database on RHEL. Mongo ports to the cloud as well. We needed data connection and transport between two Content Management System (CMS) versions, one in which the data content could be displayed from both sides from the different versions. We had to map applications and data migration into logical buckets — applications and database by lines of business, technical complexity, and similarity. The data rebuild took place in a contained environment and then patched into the live system to receive the data feed. This involved a micro-plan

CLOUD MIGRATION ASSESSMENT AND PLANNING

We had to port each application across two fields of operation, the first being RHEL and the other being GoCloud.  
We had to engage stakeholders, catalog resources and software on premises, examine Service Level Agreements (SLA), ensure continuous operation as we ported to the cloud.

We categorized application needs and workload, for example, platform, data, connectivity, security and compliance, network, high availability and maintenance.

We planned the migration to PaaS/SaaS: the *lift and shift*, application evaluation, application architecture changes, organizational readiness, for example, resources, risk, dependencies, and things like this.

At several stages of the planning we had to rollback due to incorrect assumptions. What emerged as eminent in the process was sequence, like the chess game analogy. If the command line was integral to the functioning of an application, then environment has to account for it in its planned migration before we started application migration. What parts get decommissioned first? What is the impact on the overall system? What would create a service outage? What would violate the IT Department's SLA with the Ministry? The rollbacks in the plan were part of the experimental development, the refinement that brought a cogent plan into focus.

PLAN SPRINTS TO MIGRATE

We used sprints to define milestones and safe places from which to deploy the next movement in the plan.

For cloud build and implementation support we prepared migration plans and executed iterative sprints; we selected applications and strategy — re-host, refactor, new architecture, rebuild.

We applied and implemented the migration strategy by sprints to move from on-premises to Azure ™with automation tools;  
we optimized the sprints, improved capabilities and analyzed cost, risk, and the like; we secured and managed migration sprints, Security, Data,  
protection, network access,  
control, NSG Azure DDoS protection, and others.

DELIVERY TEAM AUTOMATION

Part of the plan per the original RFP was coaching and knowledge transfer.  
We had to engage stakeholders and Ministry team members; prepare and communicate assessment reports and migration plans on a spec by spec, part by part basis;  
we had to analyze and evaluate dependencies before starting iterative migration sprints.

We used systematic investigation as our starting point. As more data became available tweaks were made to further refine it. On gross miscalculation, after thorough consultation with the Ministry, the process rolled back to its last working link. The rollbacks formed the basis of the learning and development of a thorough process for double port migrations like UNIX to Linux to GoCloud.

### What scientific or technological advancements did you achieve or attempt to achieve as a result of the work described in line 244? (Maximum 350 words)

We created a detailed process to move three layer systems from a legacy infrastructure to the cloud. We thoroughly documented the stages from UNIX/IBM/AIX to RHEL and then to GoCloud for cloud deployment and maintenance.

We wrote our reusable business plan in strict conformity with *GO-ITS 25.21 Cloud First Principles and Security Requirements*, The Ontario Public Service (OPS) security and compliance standard.

We proved that RHEL is the best Linux platform for the level of support the Ministry requires and ease of use, making it our go-to solution for cloud migrations from UNIX variants.

The plan we created includes new ITS GoCloud infrastructure, decommissioning existing Common Cluster Platform (CCP), adopting Cloud technology, performance tuning, and setting up a Disaster Recovery Plan (DRP) inclusive of infrastructure. We wrote and planned the DRP, in terms of infrastructure and responsiveness, in line with a mission critical application in the financial sector. We sequentially migrated applications, data, conceptually ported to the cloud, developed sprints to migrate with no application interruption between legacy and cloud.  
We automated delivery team automation.

We developed developed the scripts to stand up an environment, migrate data and applications in a repeatable and configurable way. We provided invaluable knowledge and experience to the Ministry, which they can now share with the Ontario Public Sector for more defined Requests for Proposals (RFPs) from vendors. We have also given ourselves a marked advantage in this marketplace because we can respond to RFPs with more precision and acumen than any vendor who has not realized a UNIX to cloud legacy migration.