DevOps Case Study

BACKGROUND

A North American IT organization was structured into 9 service delivery teams. Each of these teams were focused on a different application (email, finance, etc.) or technology (networking, data centers, security). These divisions were arbitrary and isolated; the Project Manager's (PM) were responsible for figuring out which subset of teams were needed for each project.
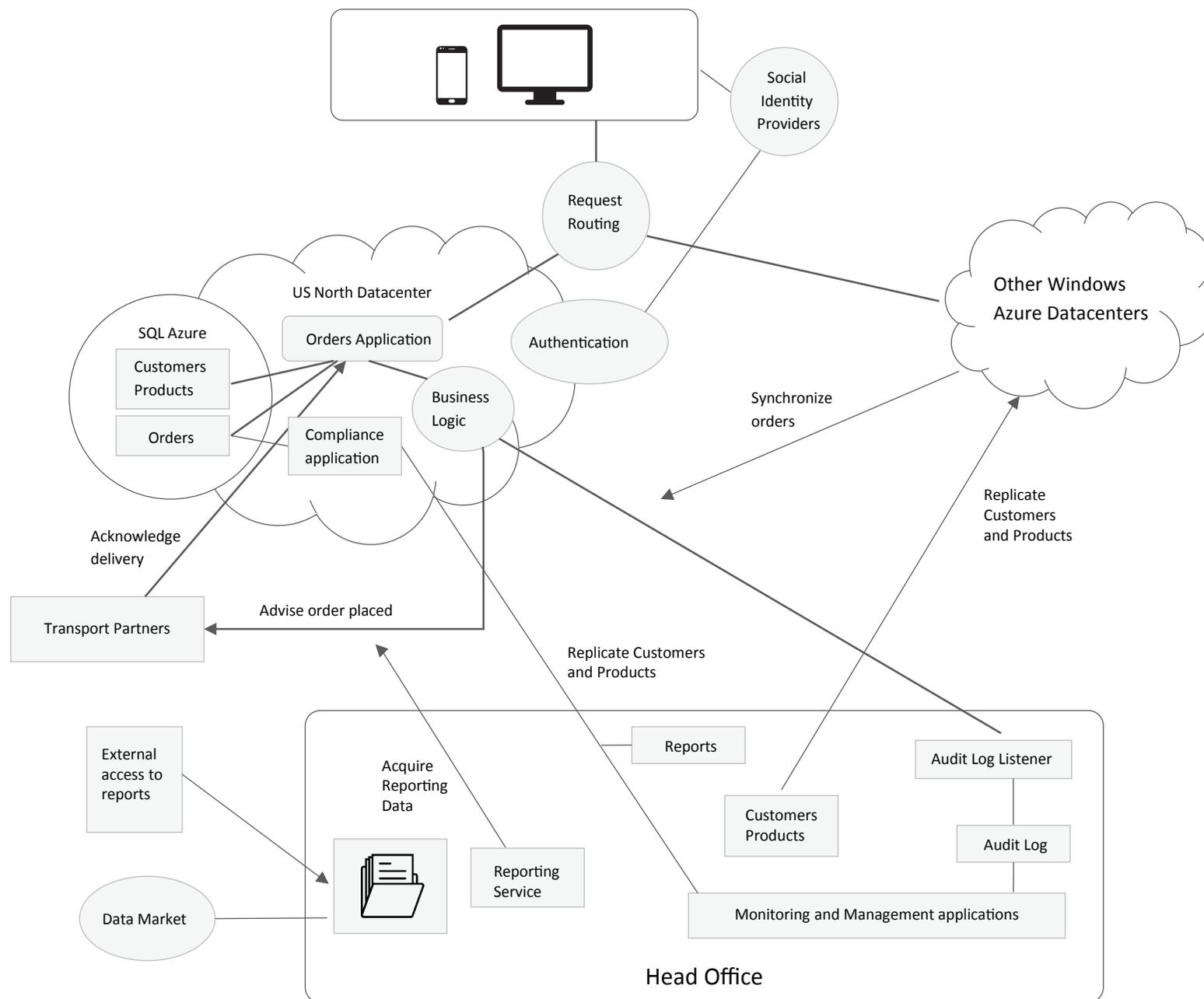
The process for each project was waterfall. It took a minimum of 9 months to gather the requirements, design, develop, and finally deploy the result. This waterfall was enforced through gate review meetings and a strong change control process, which required prior gates to have occurred before updates could flow between phases (e.g., before code could be moved from Development to Stage or Stage to Production).

The Cloud Brings Change

The appearance of Amazon Web Services (AWS) meant that the IT division had competition for the first time in their history. All the teams, considered AWS to be a threat, how it was a direct threat to the Datacenter team. Even if clients worked around the IT department by using AWS, they still required the services of the Security, Network, Development, and other teams.

mobinius ™
doing IT better

Implementation Diagram

Social Identity Providers

Request Routing

Other Windows Azure Datacenters

US North Datacenter

SQL Azure

Customers Products

Orders

Orders Application

Compliance application

Authentication

Business Logic

Synchronize orders

Replicate Customers and Products

Acknowledge delivery

Transport Partners

Advise order placed

Replicate Customers and Products

External access to reports

Acquire Reporting Data

Reports

Audit Log Listener

Data Market

Reporting Service

Customers Products

Audit Log

Monitoring and Management applications

Head Office

mobinius ™
doing IT better

Phase 1 Implementation

The Project Manager desired to deliver services faster by adopting DevOps techniques, such as CI/CD, and new technologies, such as AWS, which meant getting all the various teams on board to launch their first project on AWS.

The Front End developers and middle-tier support team were enthusiastic about being involved, since it was AWS. Since, we all know when a new technology or a tool is being introduced, the acceptance is gradual. In this particular case, the leader of the Datacenter team actively tried to sabotage DevOps effort. He was the acting lead of the Network team, which gave him additional leverage as AWS required that team's VPN and network services support to extend the domain to the new cloud environment. Also, he was the owner/co-chair of the change control board and, prior to the introduction of AWS, had full authority to accept or deny any system or project being installed in the datacenter.

The sabotage and other political moves created an "Us-versus-them" situation. People were picking sides, often based on who they thought would "win" rather than what was best for the organization.

The PM ultimately realized he was fighting against insurmountable headwinds and that this was not the way to win the hearts and minds of those involved. A new strategy was needed.

mobinius™
doing IT better

Phase 2 DevOps comes as a successor

The final attempt at adopting DevOps principles involved a strategy of consolidating passion, skills, and willingness, all roped into one team. The PM hoped to leverage the success of this team to encourage the spread of new ideas and ways of working. Also, this time the PM was careful enough to choose people wisely for the new team.

Their first project was estimated to take around 8 months. However, the CIO was firm in completing this project in 4months itself. The team lead the migration of their portfolio of apps. They used tools that were not approved by the central IT organization, but it was easier to beg forgiveness than get permission. This included using GitHub Enterprise, an open source Agile backlog tool, and AWS.

They were radically successful. After a six-month lead-in period, the team started doing weekly incremental deployments to AWS. They were able to launch in four months and stayed within budget. They also experienced a high profile success during the project when there was a news event that created the need for an ad-hoc website update. The DevOps team was able to rearrange their priorities and respond to this feature request in under a week and, once tested, deploy the update in 24 hours to the website using their automation tools. They were also able to quickly engineer the website to scale elastically to support the high-traffic, high-profile announcement.

mobinius™
doing IT better

Hindsight Is How We Learn

We are currently working on Way forward – making it happen

- Transforming a new team every few months rather than waiting for the DevOps experiment to be complete. This would built a community of expertise that will snowball into a larger, more sustainable success.
- DevOps team partnering with stakeholders faster by including them in daily stand-ups, involving them in Scrum or Kanban activities, and so on.
- Focus on rewarding not just the first team, but all teams. The first team got the top cover and did it out of love. The other teams took more personal risk because they weren't "true believers."
- Mandate all new projects in the organization to start with DevOps practices.

mobinius™
doing IT better