



# Digital Business Architecture

Best practices for planning, managing  
and delivering Digital Transformation

# Table of Contents

1. Transformational Blueprints for Digital Disruption
2. Blockchain and the CIO: A new model for IT
3. Netflix - Exemplar blueprint for Cloud Native Computing
4. Vodafone and Netflix - Reference blueprints for Platform Business Models
5. Agile Business Architecture
6. Microsoft: Cloud Solution Design best practices
7. DevOps and Digital Transformation Roadmaps
8. About The Author

# Transformational Blueprints for Digital Disruption



Disrupt or be disrupted. That is the simple choice facing every CEO today, whether they will have their own "Kodak Moment" and be decimated by a digital native competitor.

Even just 10 or 20 years ago they were significant threats, but were only '800lb gorillas'.

Now they are 8,000lb gorillas, competitors who move and adapt at the speed of the new startups ventures that they are, without the legacy baggage holding them back, but who are also funded to eye watering levels like **\$8.8 BILLION!**

Few corporations let alone startups can leverage that kind of capital, and the fact they combine it with ultra lean approaches to technology and business models means they can concentrate these monies on global marketing and expansion, at the expense of their competitors, ie potentially **your** industry next.

Therefore as many management experts point out organizations that have a more traditional, cost-management and operational view of technology, need to rapidly become far more strategic.

HBR suggests the CIO could become the [Chief Digital Officer](#) to better lead the use of technology for strategic customer programs, and McKinsey says they need to lead as the '[Transformer in Chief](#)'.

## Web Scale IT

In a VentureBeat article the author envisions '[the future of enterprise tech](#)', describing how pioneering organizations like Netflix are entirely embracing a Cloud paradigm for their business, moving away from the traditional approach of owning and operating your own data centre populated by EMC, Oracle and VMware.



Instead they are moving to 'web scale IT' via on demand rental of containers, commodity hardware and NoSQL databases, but critically it's not just about swapping out the infrastructure components.

Critically this isn't just about scaling, although obviously for businesses expanding exponentially it is essential.

What's more important is the rapid service innovation it enables. If your business systems are rigid and hard to change then obviously launching the next market dominating digital service will be slower and more expensive for you.



All kinds of small as well as large features can be highly impactful. For example offering new contact channels with customers via Twitter, [calling them via a Tweet](#) or enabling self-service access using hashtags like [O2](#) and [Tweetserve](#).

[Starbucks in particular](#) is notable for great success with this type of innovation, such as growing customer loyalty and repeat business through their [Tweet a Coffee](#) app.

## Cloud Native, Digital Native

Cloud computing is the core foundation to this agility, a special blessing for startups with zero legacy baggage.

For example as [Diginomica writes](#) Mondo has launched the first 'Cloud Native Bank' - A digital bank born and enabled entirely on Cloud computing. This is part of a trend that recognizes how hyper-competitive 'digital native' businesses can be, with [Starling also launching](#) as a digital only banking service.

However it doesn't mean it's not accessible to long standing corporations who do have massive estates of COBOL mainframes et al. It does require a deep transformational exercise to migrate and modernize software, but once achieved that delivers a platform capable of this much higher frequency digital innovation.

For example although considered a digital native it's often forgotten that it wasn't that long ago Netflix was as old fashioned as Kodak too.

What [this case study](#) demonstrates is how they took the bull by the horns and transitioned all of that legacy IT to AWS, modernizing it as they went, to the extent they have become the uber poster child for the Cloud Native trend.

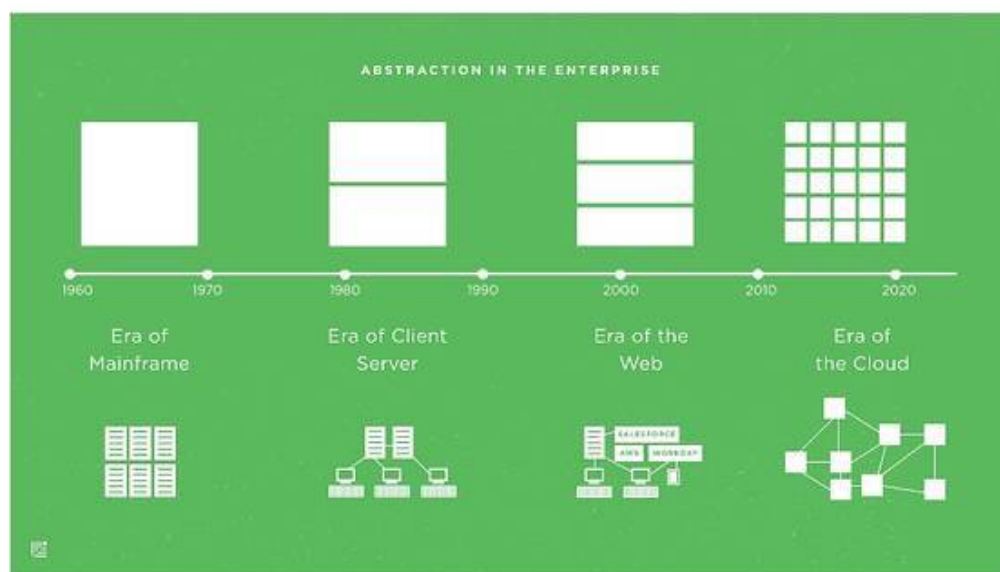
Now they are able to rapidly launch multiple new services globally, like [pioneering entirely new movie formats](#), and we all know how well that has turned out for them.



# From Mainframe to Microservices - Harnessing the Cloud Era

Of course the primary context for this evolution is the competitive advantage it makes possible, the ability to 'disrupt' industries through much faster and more effective innovation cycles.

A great primer for Microservices, one of the core Cloud Native principles, is the Wall Street Journal article [Innovate or Die - The Rise of Microservices](#), where they contextualize its emergence as part of the fourth wave of IT, the **Era of the Cloud**.



The article very eloquently maps out a timeline of the ongoing evolution of enterprise software architecture, from 'Mainframe to Microservices', the modular software design approach that Cloud enables, and would place these leaders at the far right, versus the enterprise incumbents who are typically spread across the whole timeline, right back to the mainframes that still run giant business exchanges like credit card payments.

The opening line sums up the inflection point we are experiencing:

*"Software has emerged as the critical differentiator in every industry, from financial services to fashion, as "technology first" startups disrupt global markets."*

In other words the central theme is that software engineering is emphasized

as a method of strategic advantage, by 'technology first' businesses, it's no longer just a necessary operational cost. Indeed [research](#) has shown that those firms that invest more in new innovation-centric software techniques like DevOps and Continuous Deployment, are generating more revenues than their 'laggard' peers.

## Disruptive trends

Of course tech alone doesn't make for a business model; rapidly creating new software that users don't value will be equally unsuccessful, as will business models that don't scratch a market itch in a new way.

Therefore creative strategy that leverages the technology is still just as if not more important, and a great way to stimulate this process is to consider global trends and how they might impact your industry, and be the first to deliver that impact.

## The On Demand Economy

These are dynamics that can be reused by others the same way, for example Uber Taxis and Airbnb both exploit the new 'On Demand Economy'.

[This Business Insider article](#) describes the On Demand Economy as the "uberfication of the service industry", how this is made possible by an On Demand business framework of core services, cloud applications, and ubiquitous consumer technologies combined with their appetite for instant gratification via these devices.

It defines that On Demand is *"the economic activity created by technology companies that fulfill consumer demand via the immediate provisioning of goods and services"*, and that this is achieved via overlaying a 'digital mesh' across existing infrastructure networks to harness available supply resources so that they can be hired at the tap of a smartphone button.

Another Economist article [There's An App for That](#) covers how the model is rapidly expanding into numerous industries, citing numerous Uber-like startups repeating the success for markets like cleaning, food delivery and home handyman services.

## Google Maps as Social Market Networks

An especially powerful approach is to zero in on the intersection of trends.

For example in [this TechCrunch article](#), the author [@JamesCurrier](#) describes this effect, how new ventures are blending social networks with e-marketplaces and the On Demand Economy, where they:

- *"Combine the main elements of both networks and marketplaces*
- *Use SaaS workflow software to focus action around longer-term projects, not just a quick transaction*
- *Promote the service provider as a differentiated individual, helping to build long-term relationships"*

These are tremendously powerful insights, and when you also further consider the idea that [Google Maps are becoming social networks](#), the potency of this ultra niche segments is especially tantalizing.

In short the key is to not see these trends as isolated business models and technologies, like Social Networks vs Maps, but rather that these are attributes that any business model can be assembled into a new, unique and devastating way of serving customer needs.





# Blockchain and the CIO: A new model for IT

Guest contribution from Don and Alex Tapscott

**The most important emerging technology for the enterprise – and therefore the CIO – is not big data, the social web, artificial intelligence, robotics or the cloud – it's blockchain.**



The digital revolution is bringing a new and radically different platform for business and other institutions that can take us through the next quarter-century of human progress. At the core is the biggest innovation in computer science in a generation. It is [blockchain](#) – the technology underlying the digital currency Bitcoin. This technology platform is open and programmable.

For the last few decades, we have had the internet of information. [Blockchain is bringing the internet of value](#). As such, it has the potential to unleash countless new applications and as-yet unrealized capabilities that could transform everything in the next 25 years.

At its most basic, [blockchain is a global database](#) – an incorruptible digital ledger of economic transactions that can be programmed to record not just financial transactions, but virtually everything of value and importance to humankind: birth and death certificates, marriage licences, deeds and titles of ownership, educational degrees, financial accounts, medical procedures, insurance claims, votes, transactions between smart objects, and anything else that can be expressed in code. This ledger represents the truth because mass collaboration constantly reconciles it.

We will not need to trust each other in the traditional sense, because the new platform ensures integrity. Think about it like this: trust achieved through

clever code and mass collaboration. Collective self-interest, hard-coded into this new native digital medium for value, would ensure the safety, security and reliability of commerce online. Trust is programmed into the technology, which is why we call blockchain the “trust protocol”.

Some scholars have argued that the invention of double-entry book-keeping enabled the rise of capitalism and the nation state. Today, the new platform enables a reconciliation of digital records – call it the digital reconciliation. The “internet of everything” needs a “ledger of everything”. Business, commerce and the economy need a digital reckoning.

## **Building 21st Century Companies**

It turns out that every business, institution, government and individual can benefit in profound ways.

How about the corporation – a pillar of modern capitalism? With the rise of a global peer-to-peer platform for identity, trust, reputation and transactions, we will be able to re-engineer deep structures of the firm, for innovation and shared value creation. We are talking about building 21st century companies that look more like networks than the vertically integrated hierarchies of the industrial age. The whole [financial service industry is already being reinvented by blockchain](#), and others will soon follow.

How about the internet of things? In the not-too-distant future, billions of smart things in the physical world will be sensing, responding, communicating, sharing important data; and generating, buying and selling their own electricity, doing such things as protecting our environment, charging our homes and managing our health. It turns out that this [internet of everything needs a ledger of everything](#).

As with major paradigm shifts that preceded it, [blockchain](#) will create winners and losers. But if we do this right, blockchain technology can usher in a halcyon age of entrepreneurship, empower us to reinvent our institutions for the better and create a fairer and more prosperous world.

This creates significant [opportunities and challenges for the CIO](#). Like other big innovations, such as the PC, the web, mobility and the social web, [blockchain experimentation](#) often starts outside the IT function. Thoughtful CIOs should view this as positive because every business will become a blockchain business and every business leader needs to explore

opportunities for transformation.

The trouble is that IT challenges are enterprise challenges. Companies need to have an integrated enterprise architecture to have a single version of the truth and to harness the power of blockchains. They need to have [security standards](#) and systems to protect them from bad actors. They need backup capabilities to ensure business continuity. They need an enterprise strategy for the next generation of blockchain collaboration tools and systems to cut across business silos. They need to have elite IT talent to deal with the many complexities of becoming a blockchain business.

## **A New Model for IT**

There is a solution to this dilemma. A new model of the IT function is emerging – one that makes the CIO more important than ever.

Call it the “blockchain services supermarket”. Here’s how it works: the CIO anticipates business needs and provisions a rich supply of services, from standards for blockchain application development and architecture-compliant applications, to elite talent with expertise in blockchain architecture and development – all in the “shelves” of a supermarket.

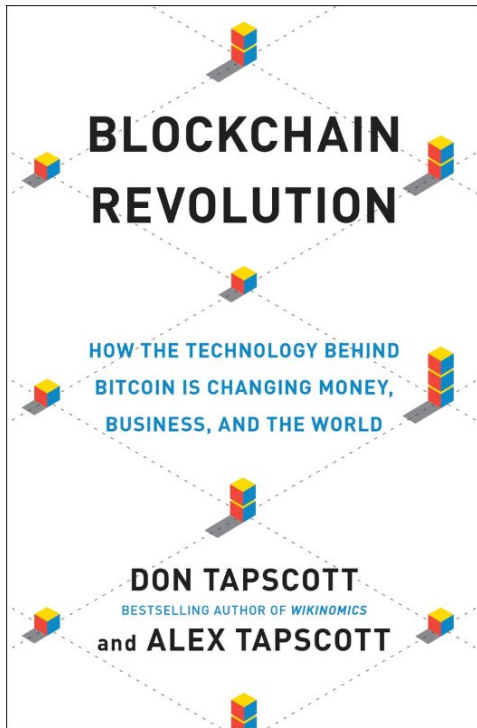
The business customer goes to the supermarket – a self-service portal or catalogue – and pulls up the available services. They choose the services, the level of service required, and combines them to meet their technology needs.

How can a CIO get started? Here are some suggestions:

- Personal use of this technology is a precondition to comprehension. Get a digital wallet on your mobile phone and buy something with digital currencies.
- Takes steps to ensure you are informed about the blockchain revolution.
- Hire or transition key blockchain IT talent to get going.
- Initiate a next-generation blockchain architecture project. Every firm will need a target architecture and a migration strategy so that new investments contribute to a desired future, rather than perpetuating the past.
- Launch a pilot where your company can learn, gain experience and make initial successes. Think big, but start small.

Start fast, too. It is likely that the second era of the internet will happen much more quickly than the first, because many of the technological preconditions

for it to take off already exist.



*Don Tapscott is the author of 15 books about the digital revolution in business and society, dating back to 1981. Alex Tapscott is the CEO of North West Passage Ventures, an advisory firm building early-stage companies in the blockchain space. Together, they are authors of [Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business and the World](#), published by Portfolio Penguin. This post was originally published in the UK in [Computer Weekly](#).*

Want to see more? Visit [Blockchain-Revolution.com](#), or [Follow Don on Twitter](#), or [Follow Alex on Twitter](#).



# Netflix - Exemplar blueprint for Cloud Native Computing

The uber poster child of migrating legacy applications and IT systems via the 'Cloud Native' approach is Netflix. Not only do they share their best practices via blogs, they also share the software they've created to make it possible via open source.



## Migrating to Web-Scale IT

In a [VentureBeat](#) article the author envisions 'the future of enterprise tech'.

They describe how pioneering organizations like Netflix are entirely embracing a Cloud paradigm for their business, moving away from the traditional approach of owning and operating your own data centre populated by EMC, Oracle and VMware.

Instead they are moving to 'web scale IT' via on demand rental of containers, commodity hardware and NoSQL databases, but critically it's not just about swapping out the infrastructure components.

## Cloud Migration Best Practices

In **this blog** they focus on the migration of the core Netflix billing systems from their own data centre to AWS, and from Oracle to a Cassandra / MySQL combination, emphasizing in particular the scale and complexity of this database migration part of the Cloud Migration journey.



This initial quote from the Netflix blog sets the scene accordingly:

*On January 4, 2016, right before Netflix expanded itself into 130 new countries, Netflix Billing infrastructure became 100% **AWS cloud-native**.*

They also reference a [previous blog](#) also describing this overall AWS journey, again quickly making the most incisive point - this time describing the primary inflection point in CIO decision making that this shift represents, a move to **'Web Scale IT'**:

*That is when we realized that we had to move away from vertically scaled single points of failure, like relational databases in our datacenter, towards highly reliable, horizontally scalable, distributed systems in the cloud.*

## Cloud Migration: Migrating Mission-critical Systems

They then go on to explain their experiences of a complex migration of highly sensitive, operational customer systems from their own data centre to AWS.

As you might imagine the core customer billing systems are the backbone of a digital delivery business like Netflix, handling everything from billing

transactions through reporting feeds for SOX compliance, and face a 'change the tyre while the car is still moving' challenge of keeping front-facing systems available and consistent to ensure unbroken service for a globally expanding audience, while conducting a background process of migrating terabytes of data from on-site enterprise databases into the AWS service.

- *We had billions of rows of data, constantly changing and composed of all the historical data since Netflix's inception in 1997. It was growing every single minute in our large shared database on Oracle. To move all this data over to AWS, we needed to first transport and synchronize the data in real time, into a double digit Terabyte RDBMS in cloud.*
- *Being a SOX system added another layer of complexity, since all the migration and tooling needed to adhere to our SOX processes.*
- *Netflix was launching in many new countries and marching towards being global soon.*
- *Billing migration needed to happen without adversely impacting other teams that were busy with their own migration and global launch milestones."*

The scope of data migration and the real-time requirements highlight the challenging nature of Cloud Migrations, and how it goes far beyond a simple lift and shift of an application from one operating environment to another.

## Database Modernization

The backbone of the challenge was how much code and data was interacting with Oracle, and so their goal was to 'disintegrate' that dependency into a services based architecture.

### ***"Moving a database needs its own strategic planning:***

*Database movement needs to be planned out while keeping the end goal in sight, or else it can go very wrong. There are many decisions to be made, from storage prediction to absorbing at least a year's worth of growth in data that translates into number of instances needed, licensing costs for both production and test environments, using RDS services vs. managing larger EC2 instances, ensuring that database architecture can address scalability, availability and reliability of data. Creating disaster recovery plan, planning minimal migration downtime possible and the list goes on. As part of this migration, we decided to migrate from licenced Oracle to open source MYSQL database running on Netflix managed EC2 instances."*

Overall this transformation scope and exercise included:

1. **APIs and Integrations** - The legacy billing systems ran via batch job updates, integrating messaging updates from services such as gift cards, and billing APIs are also fundamental to customer workflows such as signups, cancellations or address changes.
2. **Globalization** - Some of the APIs needed to be multi-region and highly available, so data was split into multiple Cassandra data stores. A data migration tool was written that transformed member billing attributes spread across many tables in oracle into a much smaller Cassandra structure.
3. **ACID** - Payment processing needed ACID transaction, and so was migrated to MySQL. Netflix worked with the AWS team to develop a multi-region, scalable architecture for their MySQL master with DRBD copy and multiple read replicas available in different regions, with tooling and alerts for MySQL instances to ensure monitoring and recovery as needed.
4. **Data / Code Purging** - To optimize how much data needed migrated, the team conducted a review with business teams to identify what data was still actually live, and from that review purged many unnecessary and obsolete data sets. As part of this housekeeping obsolete code was also identified and removed.

A headline challenge was the real-time aspect, 'changing the tyre of the moving car', migrating data to MySQL that is constantly changing. This was achieved through [Oracle GoldenGate](#), which could replicate their tables across heterogeneous databases, along with ongoing incremental changes. It took a heavy testing period of two months to complete the migration via this approach.

## Downtime Switchover

Downtime was needed for this scale of data migration, and to mitigate impact for users Netflix employed an approach of ***'decoupling user facing flows to shield customer experience from downtimes or other migration impacts'***.

All of their tooling was built around ability to migrate a country at a time and funnel traffic as needed. They worked with ecommerce and membership services to change integration in user workflows to an asynchronous model, building retry capabilities to rerun failed processing and repeat as needed.



An absolute requirement was SOX Compliance, and for this Netflix made use of components available in their OSS open source suite.

*Our Cloud deployment tool Spinnaker was enhanced to capture details of deployment and pipe events to Chronos and our Big Data Platform for auditability. We needed to enhance Cassandra client for authentication and auditable actions. We wrote new alerts using Atlas that would help us in monitoring our applications and data in the Cloud.*

## Building HA, Globally Distributed Cloud Applications with AWS

Netflix provides a detailed, repeatable best practice case study for implementing AWS Cloud services, at an extremely large scale, and so is an ideal baseline candidate for any enterprise organization considering the same types of scale challenges, especially with an emphasis on HA - High Availability.

Two Netflix presentations: [Globally Distributed Cloud Applications](#), and [From Clouds to Roots](#) provide a broad and deep review of their overall global architecture approach, in terms of exploiting AWS with the largest and most demanding of capacity and growth requirements, such as hosting tens of thousands of virtual server instances to operate the Netflix service, auto-scaling by 3k/day.

This goes into a granular level of detail of how they monitor performance, and then additionally in they focus specifically on [High Availability Architecture](#), providing a broad and deep blueprint for this scenario requirements.

## Netflix Spinnaker - Global Continuous Delivery

In short these address the two core, common requirements of enterprise organizations, their global footprint and associated application hosting and content delivery requirements, and also their own software development practices - How better can they optimize the IT and innovation processes that deploys the software systems that needs this infrastructure.

## Build Code Like Netflix - Continuous Deployment

The ideal of our 'repo guide' for the Netflix OSS suite is for it to function as a 'recipe' for others to follow, ie You too can *Build Code Like Netflix*.

Therefore it's apt one of the best starting points is their blog with the same title - [How We Build Code At Netflix](#).

Most notably because this introduces the role of [Continuous Deployment](#) best practices, and how one of their modules '[Spinnaker](#)' is central to this.

## Cloud Native Toolchain

In this blog [Global Continuous Delivery With Spinnaker](#) they explain how it addresses this scope of the code development lifecycle, across global teams, and forms the backbone of their DevOps 'toolchain', integrating with other tools such as Git, Nebula, Jenkins and Bakery.

x

As they describe:

*Spinnaker is an open source multi-cloud Continuous Delivery platform for releasing software changes with high velocity and confidence. Spinnaker is designed with pluggability in mind; the platform aims to make it easy to extend and enhance cloud deployment models.*

Their own quoted inspirations include [Jez Humble's blog and book](#) on Continuous Delivery, as well as experts such as Martin Fowler and working ideals such as '[Blue Green Deployments](#)'.

## Moving from Asgard

Their history leading up to the conception and deployment of Spinnaker is helpful reading too; previously they utilized a tool called 'Asgard', and in [Moving from Asgard](#);, describe the limitations they reached using that type of tool, and how instead they sought a new tool that could achieve:

- *"enable repeatable automated deployments captured as flexible pipelines and configurable pipeline stages*
- *provide a global view across all the environments that an application*

*passes through in its deployment pipeline*

- *offer programmatic configuration and execution via a consistent and reliable API*
- *be easy to configure, maintain, and extend"*

These requirements formed into Spinnaker and the deployment practices they describe, which you can repeat through the Github Download.

# Vodafone and Netflix - Reference blueprints for Platform Business Models



As Gartner describes APIs are at the heart of the digital business model, a topic they explore through their Gartner Platforms series, focusing on specific impacted sectors like digital banking.

This great Diginomica case study of Pearson shares insights into how one pioneering media organization harnessed APIs into their digital strategy. Pitney Bowes highlights the concept in simple terms like a 'digital supply



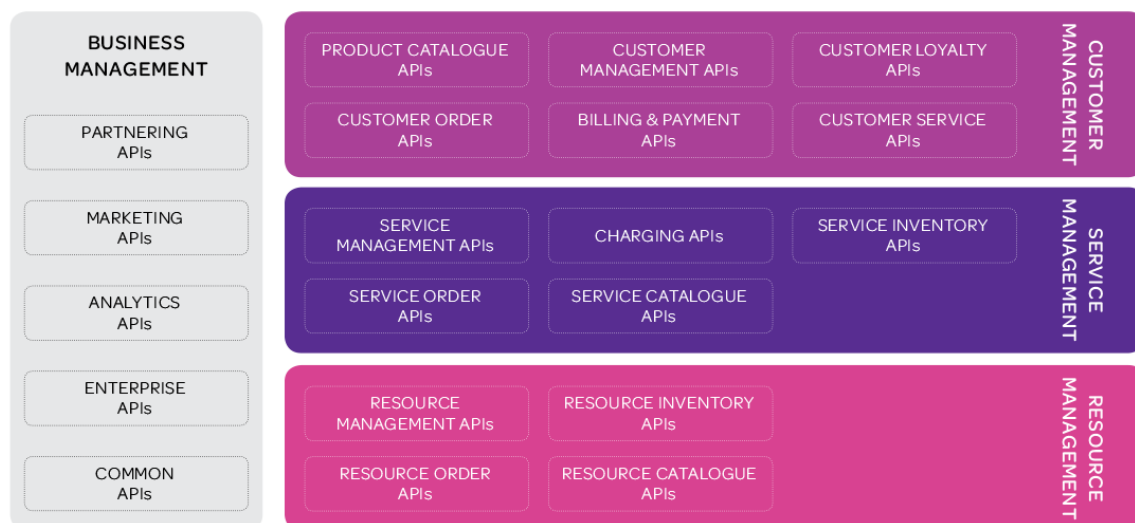
chain', how can you reflect the downstream world that sells your products, in API enabling terms?

APIs are about making the business 'programmable', like how it enabled the 'platformication' of the Evernote business model. Dr Lester Thomas, Chief Systems Architecture at Vodafone, [describes their central role](#) in developing a Telco marketplace in this form, an 'NFVaaS Ecosystem'.

## Vodafone - Platform Business Model

The role of APIs in enabling Vodafone's Platform model is described in [this TMF blog](#).

### VODAFONE'S PLATFORM ARCHITECTURE



Source: Vodafone

*Companies like BT, Orange and Vodafone see orchestration and APIs as key components of their strategies to become platform providers: Multiple orchestrators in multiple software platforms will communicate with each other and with other network and OSS/BSS components to deliver services – through open APIs.*

*They are also using the APIs to open their network to customers. "Large enterprises are demanding API access to Vodafone because*

*they want to be in control of their services, very dynamically,” says Dr. Lester Thomas, Chief Systems Architect, Vodafone Group.*

## ONAP

The other main industry trend that intersects here to very powerful effect is the [open sourcing](#) of AT&T's core NFV platform 'ECOMP', which has since accelerated into a broader industry momentum and consolidating ecosystem, with [ECOMP merging with Open-O](#) to form a new 'ONAP' - Open Network Automation Platform.

Therefore the API features of ECOMP will now be features of an open source distribution for others to use the same way, such as defined in this earlier [this TelecomTV interview](#), where AT&T's Director of Ethernet Product Management [Dan Blemings](#) describes their motivations for the disruptive ECOMP move, and specifically the role of APIs as a central feature:

- API-driven tools for wholesale providers to on board into their systems so they can provision ethernet services in real-time, versus weeks and months.
- They are opting for open standards versus proprietary vendor solutions, essential to this supplier interoperability, such as the [MEF 55 LSO Reference Architecture](#) .
- Open sourcing ECOMP is intended to provide the industry a 'shot of adrenaline' that drives compability through a faster, more innovative ecosystem, particularly encouraging contributions back to the open source to spur innovation.
- This is central to their Platform Business Model.

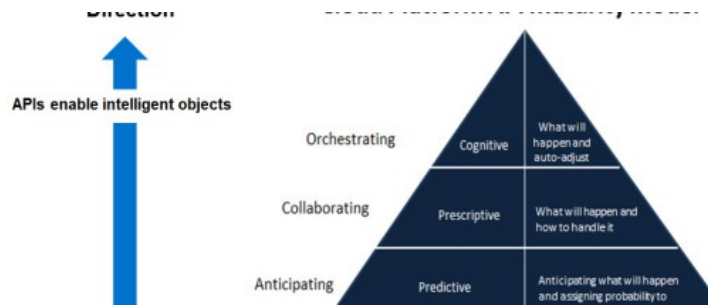
## Maturity Model - APIs and Microservices

In [this LinkedIn blog](#), Anil Madan specifically describes the role of the API in the Platform Business Model, and also suggests this API Maturity Model:

- **Basic** - exposes a basic API with standards, fully tested.
- **Self-served** - complete developer experience, available on a services portal with complete documentation and samples.
- **Performant** - Complies with Service Level Objectives - response-time and availability. This should be table stakes before APIs are exposed to 2-party and 3-party.
- **Isolated** - fully isolated, with code and data encapsulated.

- **Portable** –portable units built around a container technology that can be independently deployed ideally in a cloud.

The Forbes blog [2017 year of API Economy](#) describes an API Maturity Model:



and [CIO.com begins to explore](#) a headline theme within this maturity journey: The complimentary fusion with a microservices architecture.

## Netflix API

When [Netflix began their API strategy](#) they were still in the business of shipping DVDs. Since then of course their online global content delivery network approach has expanded exponentially, including across multiple devices as well as geographic markets.

They describe a specific philosophy blueprint for how to define such a platform:

- Embrace the Differences of the Devices
- Separate Content Gathering from Content Formatting/Delivery
- Redefine the Border Between "Client" and "Server"
- Distribute Innovation

In [2011/12](#) Netflix replanned their core API strategy, published to their [own blog](#) and also [ProgrammableWeb](#).

By 2013 they describe in [Deploying the API](#) how their API had evolved to become *"the Netflix API serves as an integration hub that connects our device UIs to a distributed network of data services"*, and shared a series of InfoQ blogs on their ongoing [API Optimization](#).

[This blog](#) brings us up to 2016, where Netflix describe their API as:

*"The Netflix API is the "front door" to the Netflix ecosystem of microservices. As requests come from devices, the API provides the logic of composing calls to all services that are required to construct a response.*

*It gathers whatever information it needs from the backend services, in whatever order needed, formats and filters the data as necessary, and returns the response."*

## Embrace the Differences of the Devices

The value of these philosophies is immediately evident when you consider the nature of the challenges they are seeking to address:

*The key driver for this redesigned API is the fact that there are a range of differences across the 800+ device types that we support. Most APIs (including the REST API that Netflix has been using since 2008) treat these devices the same, in a generic way, to make the server-side implementations more efficient.*

and how

*While effective, the problem with the OSFA approach is that **its emphasis is to make it convenient for the API provider, not the API consumer.***

A market explosion for Netflix has meant they've expanded across multiple strategic dimensions simultaneously: A shift from DVD to online platform, a massive surge in traffic through moving online, an expansion to multiple countries across the world and an enormous expansion of capacity and devices: Broadband, iPads and smartphones have all contributed to an exponential growth factor.

It's the multiple formats of these devices that causes the challenges. Netflix wants to exploit the rich features of each but REST isn't intended for that granularity; it achieves ubiquity through simple operations only. Therefore to enable exploitation of these rich features for their own content strategy ambitions, they expanded their approach.



In [this ProgrammableWeb guest blog](#) Daniel Jacobson, Director of Engineering for the Netflix API, provides a much more detailed analysis explaining this rationale.

Netflix define that the OSFA approach of REST for a Digital strategy of this magnitude can prove inadequate in these terms and is thus a key insight for others planning a similar scale and mode of platform strategy.

## Implementation

They also share their crucial insights learned from implementing the approach, most notably the act of delegating ownership of APIs to the device teams.

As you might imagine the principle challenge that would arise from this type of approach is API complexity, with so many in use. Netflix scales to the challenge through distributing the workload across the end point development teams:

*As described above, pushing some of the client code back to the servers and providing custom endpoints gives us the opportunity to **distribute the API development to the UI teams**. We are able to do this because the consumers of this private API are the Netflix UI and device teams. Given that the UI teams can create and modify their own adapter code (potentially without any intervention or involvement from the API team), they can be much more nimble in their development.*

offering architecture benefits:

*because these adapters are isolated from each other, this approach also diminishes the risk of harming other device implementations with tactical changes in their device-specific APIs.*

but presenting these fundamental challenge:

*Of course, one drawback to this is that UI teams are often more*

*skilled in technologies like HTML5, CSS3, JavaScript, etc. In this system, they now need to learn server-side technologies and techniques.*

Netflix describe how they address this challenge simply through always recruiting skilled engineers who can cope with these development requirements.

# Agile Business Architecture

Bringing Agile to the Whole Organization



DevOps only accelerates one part of a business, software engineering.

It doesn't speed up how quickly sales teams can turn around proposals to new clients or reduce wait times for Healthcare teams.

But it can. The same working practices can be generalized and applied to any organizational team for the same benefits, an effect HBR calls '[Bringing Agile to the Whole Organization](#)', and that Mastercard offers repeatable best practices for.



As the title captures the core purpose is to generalize the specific working practices of software developers, and reuse them across the entire organization, not just IT. It asks the question how can the overall enterprise become more agile, by widespread adoption of enterprise best practices for collaboration and product delivery.

## Agile Business Architecture

Mastercard offers a set of presentations defining Agile Business Architecture, repeatable best practices that can be used to deliver this heightened high

performance.

1. [Agile Business Architecture](#) - Experiences Linking Business Architecture with an Agile/Lean Method.
2. [Business Driven Roadmaps](#) - Managing the work through feature release planning roadmaps.
3. [Architecture-Driven Investments](#) - Central governance based on Enterprise Architecture standards.

## BA User Stories

The first presentation describes the central premise, expanding User Stories to add Business Architecture context.

[User Stories](#) are the lifeblood of product and software development, and form the base unit of work for an [Agile process](#).

They can be written in a simple form on paper notes, through to being described as complex [UML diagrams](#).

Roman Pichler provides a [10 tips guide](#) for how to write them, and vendor tools can assist greatly with the process. For example Visual Paradigm offers this introductory article - [How to Write Effective Use Cases](#). They also explain [How to Map Business Processes to User Stories](#), and their product can be used for:

- [Creation of stories](#) as UML use cases
- Storyboard and team collaboration
- Easily produce a report detailing their Technical Requirements

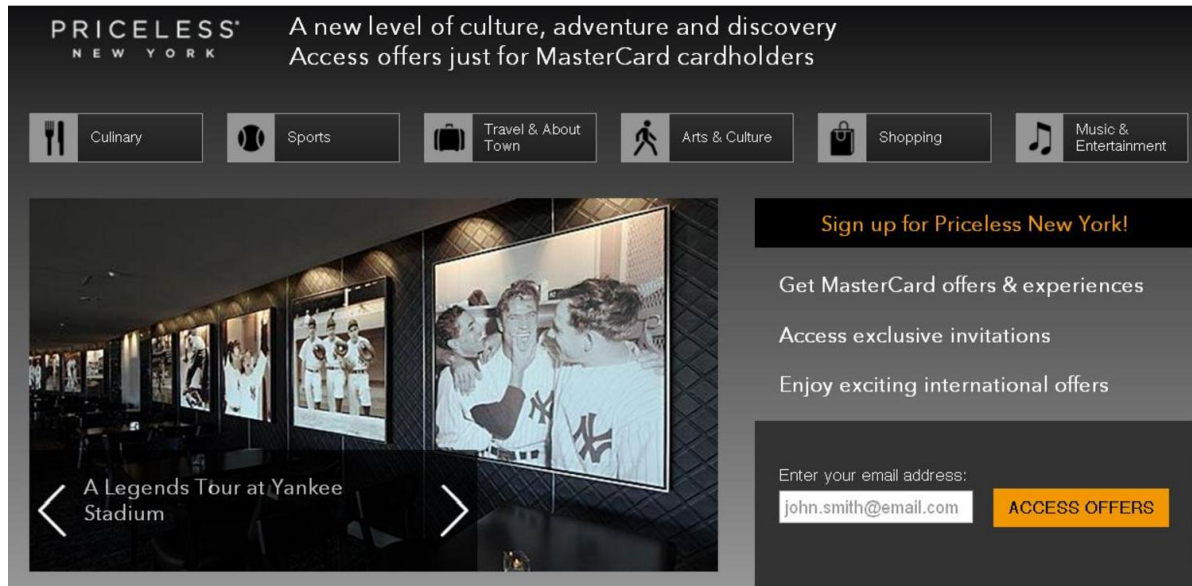


## BA 'Fragments'

Mastercard describes their use of standard agile practices, how their headline 'Product Vision' feeds the Product Backlog which in turn is fed into Sprints of 24hr and 1-4 week cycles, and how these are then augmented with what they call 'fragments of Business Architecture', with the slides below highlighting how they:

1) Encompass UI design elements, like web site widgets and updates.

## The intersection of BizArch and Agile



March 18, 2013  
Page 13

2) Describes each fragment as a component piece of the overall Business Architecture, linking them to Capabilities like 'Campaign Management'.

## A fragment of the Business Architecture

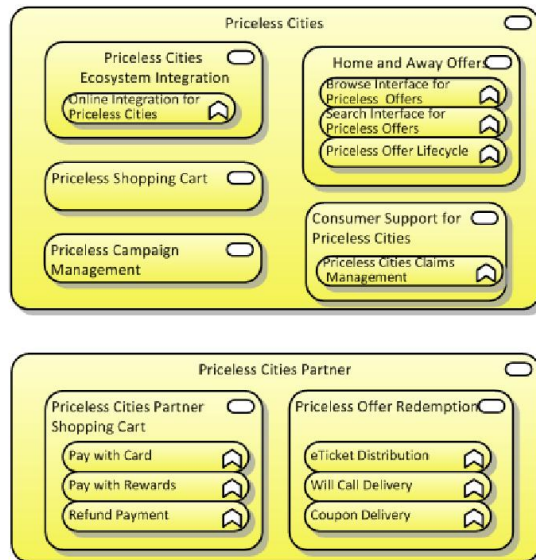
---



**3) Integrates these Capabilities into an overall Solution Architecture.**



## A fragment of the Solution Architecture



4) Organizes them all into a Features Backlog, applying an indexing references to each requirement. Mastercard uses examples like AE-79: Develop process to manage servers.

## A fragment of the Agile backlog

Issue Type	Key	Summary
Epic	AE-24	Rewards Redeem Analysis
Epic	AE-23	Rewards Balance Analysis
Epic	AE-17	SSO Issuer to MC
Epic	AE-11	Registration
Story	AE-90	Knowledge of Priceless Cities platform and road map
Story	AE-89	Knowledge of Priceless Cities templates and themes
Story	AE-86	Process for Priceless Cities Updates
Story	AE-85	MasterCard Standards for Up-time
Story	AE-84	Load/Capacity Process
Story	AE-83	Browser Support
Story	AE-82	Maintenance Windows
Story	AE-81	Monthly Analytics Reports
Story	AE-80	Include Analytics Tags
Story	AE-79	Process to Monitor Server
Story	AE-73	Transactional Emails
Story	AE-72	Offer Reminders
Story	AE-71	General Public Offers
Story	AE-70	Ability to Purchase Offers
Story	AE-69	Display Home and Away Offers
Story	AE-63	Admin Support
Story	AE-62	Target Offers
Story	AE-61	Home and Away Offers

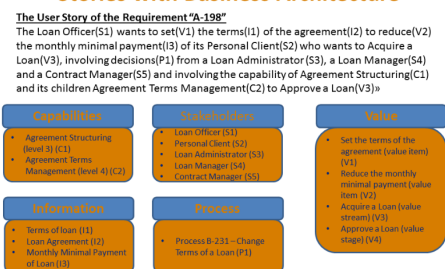
- Meta Story
- New Actor
- Match

March 18, 2013  
Page 16

Daniel Lambert also provides an [excellent article exploring this approach](#) highlighting:

- A new business process can be defined as a BA User Story such that it models all of the impacted Capabilities, and thus involve the related teams in the transformation.
- Business Architecture enables clearer requirements definition and validation, framing them based on business strategies.
- The approach avoids requirement duplications across business units and departments.

**Figure 5 - Avoiding Confusion in User Stories with Business Architecture**



Benchmark Consulting Canada  
Copyright 2013 - All Rights Reserved

## Business Driven Roadmaps

The second and third presentations are ideally considered together, offering a closed loop to link feature development right back up through the highest levels of investment decision making.

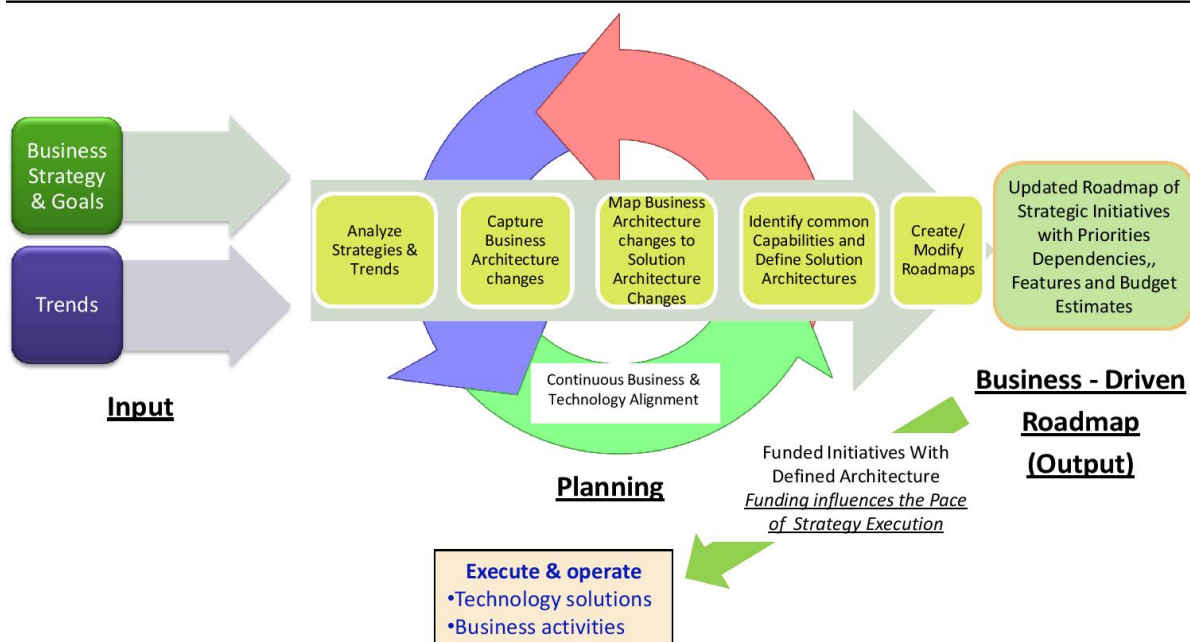
Projects are managed through 'Business Driven Roadmaps', enabling an

overall **Architecture-Driven Investment Planning** approach.

It takes inputs of Business Strategy and Goals, and market Trends, and updates and outputs a Business Driven Roadmap which details Strategic Initiatives with Priorities, Dependencies, Features and Budget Estimates. This is populated by a lifecycle of activities including:

- Analyze Strategies and Trends
- Capture Business Architecture changes
- Map Business Architecture changes to Solution Architecture changes

## Business Centered Architecture Planning and Business-Driven Roadmaps



## Solution Lifecycle Model

This planning framework is complimented by a lifecycle design process that incorporates Business Architecture, Solution Architecture and Technical Architectures, so roadmap planning is mapped to each level accordingly.

# Microsoft: Cloud Solution Design best practices

Cloud Solution Design is the review and planning process to translate customer requirements into deployed Cloud services that meet those needs.

In [this presentation](#) for building a Cloud strategy the Microsoft Enterprise Architecture and Strategy team shared their blueprint for an overall framework to guide Cloud Migration. Most notably they approach this from an Enterprise Architecture-led point of view, including encapsulating Business Architecture best practices too.

Presentation summary:

- How EA/BA provides the connecting logic between Strategy and Execution
- Mapping Strategic Goals on to relevant XaaS Cloud Services, through '**Cloud Pattern Matching**' (slide 50)
- Enterprise Architecture to Drive Cloud Strategy and Planning
- Business Value Driven Methodology
- slide 20 - connecting business strategy to technology investments
- Embraces and extends proven Cloud best practices, from NIST, Cloud Security Alliance, et al.

## Cloud Pattern Matching

Cloud Solution Design can mostly simply be thought of as a process of "XaaS Mapping", referring to identifying which particular type of Cloud service might be the best fit for your particular business requirement(s).

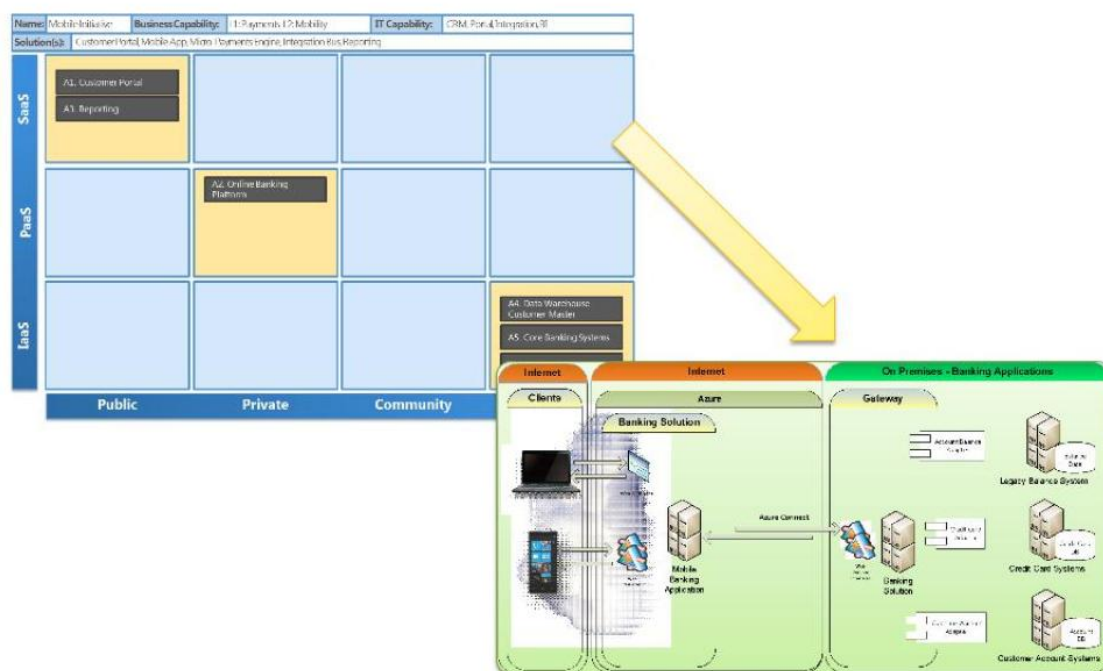
In the Microsoft presentation on slide 50 they show this XaaS Mapping, which they describe as 'Cloud Pattern Matching'.

## Example: Cloud Pattern Matching

Name:		Mobile Initiative	Business Capability:		L1: Payments L2: Mobility	IT Capability:		CRM, Portal, Integration, BI
Solution(s):		Customer Portal, Mobile App, Micro-Payments Engine, Integration Bus, Reporting						
SaaS	Public	A1. Customer Portal A3. Reporting						
PaaS	Private	A2. Online Banking Platform						
IaaS	Community							
Traditional				A4. Data Warehouse - Customer Master A5. Core Banking Systems A7. Micro-Payment Engine				
		Public	Private	Community	Traditional			

The customer requirement, eg 'A1: Customer Portal', is matched to a Public SaaS option.

## Example: Cloud Pattern Matching



This can be applied right through to scenarios including private, in-house deployments, with a common function of each mapping being the exposure and linkage to the underlying enabling technologies in a uniform fashion, across Cloud, SaaS and internal apps.

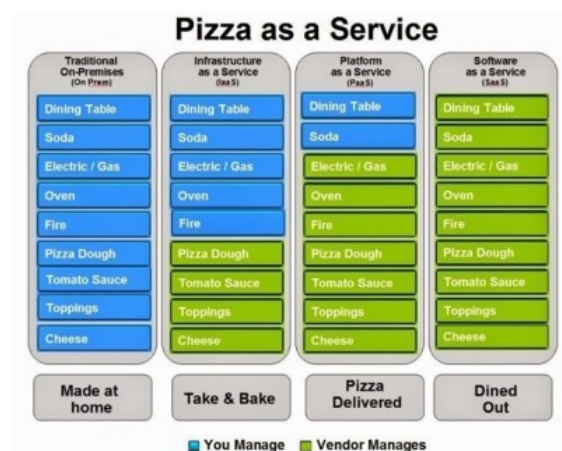
## XaaS Mapping

This top down Capability Mapping can be met by a bottom up cataloguing of potential vendor options to meet your needs, achieved in a super simplistic form through the "[Pizza as a Service](#)" analogy, where it compares the increasing scope of outsourced supplier services to how much work you do in

the preparation of the pizza dinner.

[Siemie Engineering](#) describe an equally simple example of how different vendors offer services at each level, such as:

- IaaS: Amazon AWS
- PaaS: Microsoft Azure
- SaaS: Salesforce.com



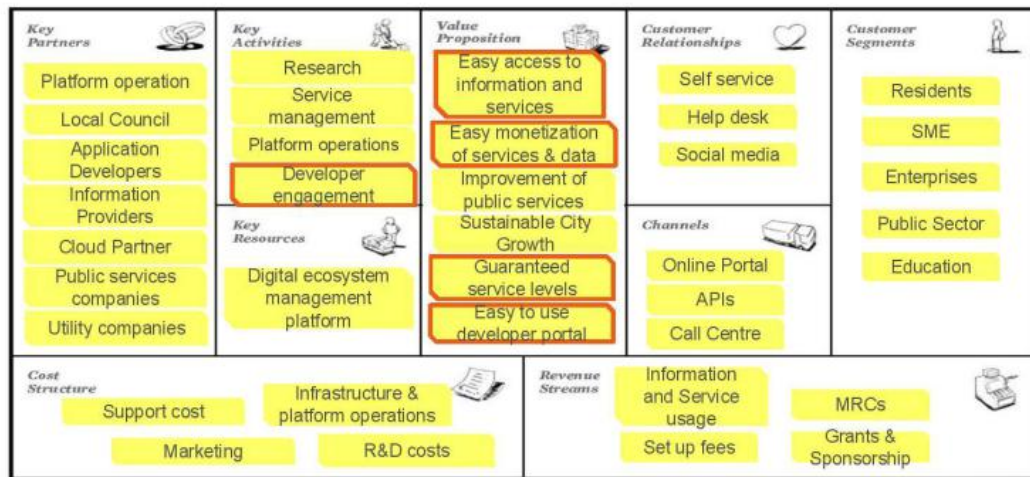
Each of these vendors also offers services in the other categories, and there are **thousands** of other options that also populate this landscape.

Your business requirements can be analyzed and mapped to one or more of these service, indicating which vendors you may want to consider for your supplier short list.

## Cloud Capability Mapping

This mapping process can be developed to a fine-grained level, very effectively demonstrated through [this presentation](#) from BT on their Milton Keynes Smart City case study, achieved through their Cloud platform (with an accompanying [executive level case study](#) from BT).



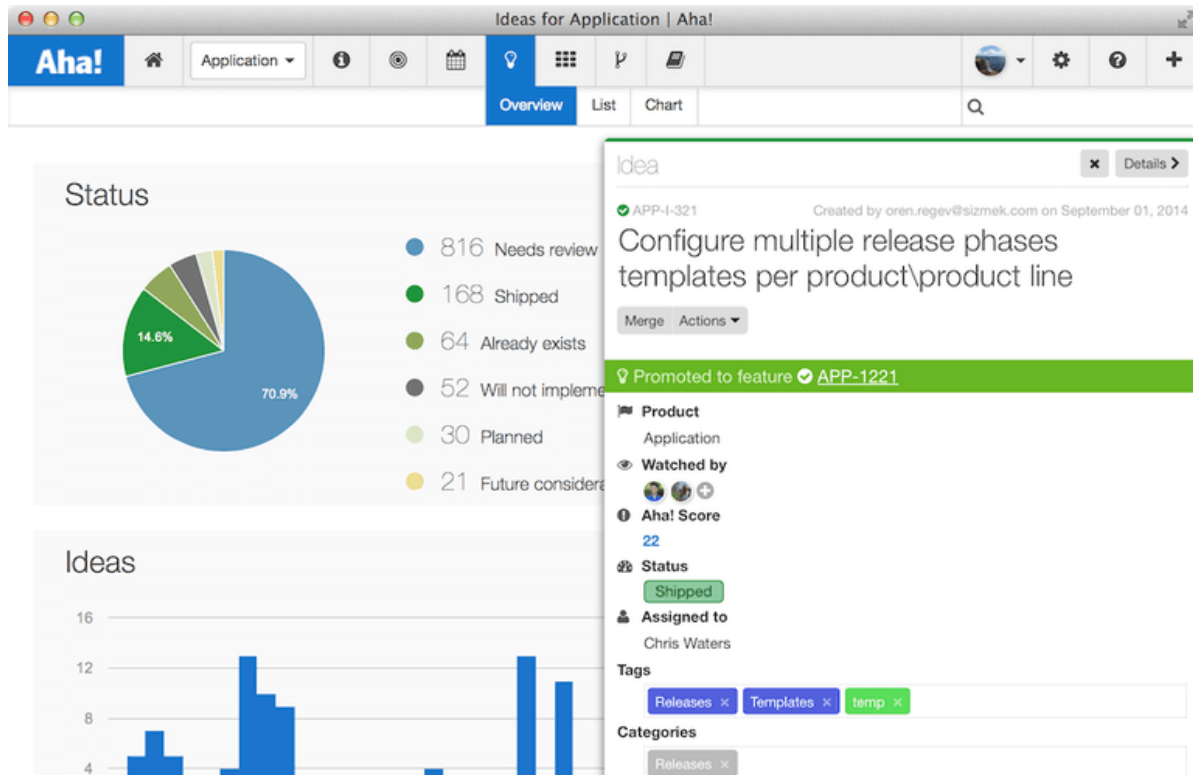


This focuses specifically on how one sub-section of the overall model is achieved, the management of SLAs for key data services.

Via the capabilities offered by Cloudsoft, BT's partner for the Cloud Management functionality built into the Platform, they isolate out and map a sub-set of the business model that the CMP can achieve, including:

- Easy access to information and services
- Easy monetization of services and data
- Guaranteed service levels
- Easy to use developer portal

# DevOps and Digital Transformation Roadmaps



DevOps alone won't deliver successful Digital Transformation.

For example it's possible to speed up software innovation but, like LinkedIn, deliver a whole bunch of features that your customers don't value and worse, pisses them off. This is far more of an issue for most organizations than their lack of IT development skills.

As [this IT Business article](#) describes businesses often lack a cohesive roadmap to set out where they are headed or how they plan to get there, exacerbating a situation of also lacking CEO and senior executive Digital leadership.

Managing accelerated innovation through PLM

# Roadmaps

To provide executives with a complete framework to manage and deliver transformation, DevOps is best combined with PLM ([Product Lifecycle Management](#)) implemented via Roadmapping apps, like [Aha.io](#) or [Uservoice](#). These provide the tools and best practices to more intelligently connect the two together, ensuring the features you prioritize are ones that deliver the most Business Value.

Readily available as SaaS options these apps provide slick tools for capturing Requirements and organizing these into Roadmaps made up of Releases. They also [integrate with Github](#) so that a complete end-to-end product lifecycle is possible, from idea and planning through deliverables and deployment.

This integrated 'toolchain' makes possible a closed loop lifecycle for managing your Digital strategy from conception through delivery:

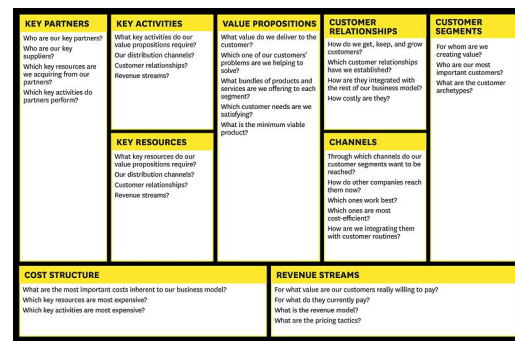
1. Establish your digital strategy, goals and product priorities.
2. Build a web portal and capture user and customer ideas and feedback.
3. Prioritize ideas into Features based on your product priorities.
4. Schedule work and produce timeline roadmap reports.
5. Synchronize roadmap with Github repo and drive work loads.
6. Report back through Agile Backlog Dashboards.

## From Canvas to Blueprint

Business Architecture offers the skills and tools to populate these tools with the details of your organizations challenges, goals and requirements, facilitating a 'Canvas to Blueprint' design process, very effectively described through presentations from Boeing and ABN AMRO.

1. **Using Business Architecture for Strategy Planning** 1) - OMG presentation from ABN AMRO.
2. **Using Business Architecture to Realize Your Target Operating Model** 2) - Business Architecture Guild presentation from Boeing.

Having sketched out your world changing new business idea on a napkin, the most popular tool for expanding it to a next level of business plan detail is the [Business Model Canvas](#).



The presentations describe how the high level Business Model Canvas defines a set of high level Strategic Goals, which are then cascaded these down through more detailed systems analysis.

They inventoried their own capabilities, scorecarding them across nine Value Propositions, to ensure they were prioritizing the right work to meet the defined Strategic Goals.

This 'idea to canvas' approach can be used in large enterprise organizations as well as startups, evolving new ventures and new product ideas from this first canvas stage to a second more detailed business model, one defined in Business Architecture terms and assets.

Enterprise organizations can work at this high level, brainstorming-centric first level, indeed it's an ideal starting point for articulating executive strategy, which can then be devolved to teams through this lifecycle process.

## Credit Suisse: Application Architecture Roadmap

Other BA best practices then provide a reference model for how these Operational Models can be translated into Cloud Application design, most notably this Credit Suisse presentation:

**'Design Reviews Using the Business Capability Model.'**

## Business/IT Alignment

This offers a core reference model for transforming IT solution delivery through Business Architecture, encompassing Business/IT Alignment, an **Application Architecture Roadmap** design review process that unifies cross-department teams and exposes duplicated functionality for consolidation.

Most notably it explicitly describes how Business Architecture bridges the gap between Business and IT, formally establishing governance links between the

goals of strategy and the execution of IT, and also how to then implement this linkage across Enterprise Architecture.

- **Slide 14** describes the core design process, a process of 'front to back' overlaying of the Logical model across the Physical one.
- **Slide 13** shows how this produces a Features List, for example T1 - A Trade Store Abstraction Layer.
- **App / Data Centre Consolidation** - This is part of a simultaneous process of identifying duplicated processes and systems, and eliminating these as part of the migration, enabled by a disposition decision tree.

This includes the required organizational Design Governance models - Credit Suisse explain how they defined one that supports the process of planning legacy migrations through a 'front to back' architecture design process that maps out the dynamics between business processes and the apps that ultimately run them.

## Application Architecture Roadmap

It provides an organizational approach for building an Application Architecture Roadmap which would drive the development and deployment of your Cloud Applications.

This process would take the required Features defined in the BA analysis process, and map these to the necessary IT service components, the Cloud Application modules and functionality.

The slides 9,10 and 14 where they show how they define the component parts of the Roadmap, and how this Feature mapping is achieved. These Features can then be fed into a DevOps pipeline and translated into the required code, ready for deployment to the Cloud.

Aha.io provides the app to manage all of these activities - For example [organizing and scheduling high volumes of Features into Epics](#) .

# About The Author

## Neil McEvoy

Neil has been a Cloud entrepreneur for over 20 years, since pioneering one of the first commercial SaaS business models in 1994 on a DEC mainframe five years before the founding of Salesforce.com, launching one of Europe's first ASPs in joint venture with Microsoft at aged 28 and co-developing numerous other startups in this one specific field since.

Most recently Neil founded the Cloud Best Practices Network, to bring together the world's leading authorities on Cloud Computing and organize a series of best practice communities to further accelerate ongoing industry innovations and help support end-user adoption through sharing open digital transformation models.

## Digital Transformation Consulting

Neil is available for consulting and specializes in the Digital Transformation best practices described in this ebook.

[Connect on LinkedIn](#) or contact via email: [neil@cbpn.io](mailto:neil@cbpn.io)