Huawei’s Future Mode of Operations

A Whitepaper

Author: Geoff Halprin, Principal Operations Architect, Strategy and Architecture, Global Technical Services

V1.0 – First Release

1 Executive Summary

IT and CT are simultaneously merging and undergoing their most significant transformation in decades – a digital business transformation. This will impact every aspect of their operations, and those of their partners, such as Huawei.

Every major CT organization has commenced their transformation journey. Huawei is positioning itself as a key ICT transformation partner. This paper introduces the concepts of the digital ecosystem, and the nature of the transformation journey that must be undertaken.

This journey is not optional. Digital companies have seen a 200-fold improvement in time to value, and a 168-fold improvement in time to recover from incidents. For CT organizations, they must evolve to compete with better service agility and the ability to define and evolve complex cross-functional service offerings. They must merge their silos and re-engineer their processes to embrace iterative, incremental, rapid delivery of value through practices such as continuous integration, continuous delivery and joint agile delivery.

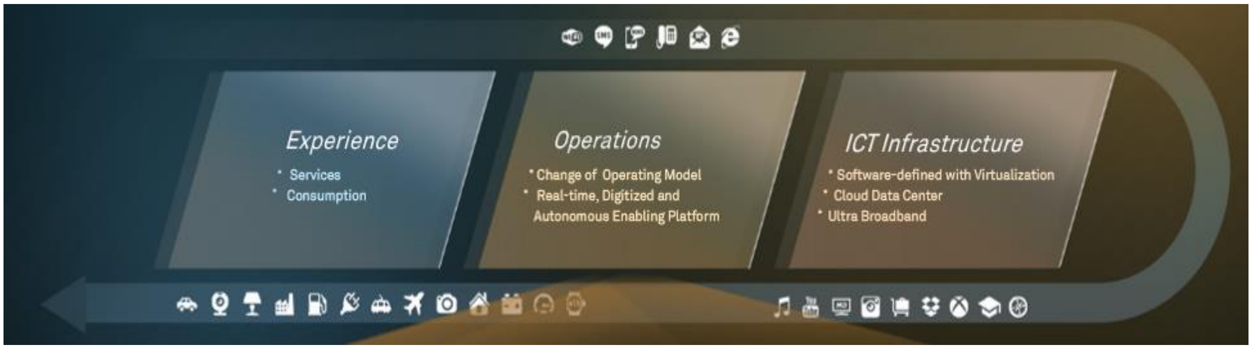
Digital companies have seen a 200-fold improvement in time to value, and a 168-fold improvement in time to recover from incidents!

Huawei must also embrace these practices if it is to be seen as a credible consulting partner. We have already seen significant interest from key customers, such as Telefonica Digital, Orange, Telenor, and Celcom on this topic, and how Huawei can assist.

Huawei recognizes the importance of this transformation. Eric Xu (rotating CEO) identified this as one of the company’s top 4 priorities:

> *Re-architect IT systems: Operators cannot compete with OTT with an as-is IT system. Huawei needs to provide a ROADS experience driven operations system (OSS+BSS), realizing Internet industry O&M and cost reductions.*

The digital business transformation is a customer experience led transformation:


Experience
• Services
• Consumption
Operations
• Change of Operating Model
• Real-time, Digitized and Autonomous Enabling Platform
ICT Infrastructure
• Software-defined with Virtualization
• Cloud Data Center
• Ultra Broadband

This feedback loop embodies the customer experience led "outside-in" nature of digital services that delight and hold customers to a brand. A new mode of operations is at the heart of this transformation, mapping these experiences to changes in services, processes and infrastructure.

2 THE DIGITAL CHALLENGE

The ICT world has changed forever. Digital innovations, driven by technologies such as Cloud, network virtualization, APIs, and microservices, have forever changed the way not only that software and infrastructure is built, but how we evolve it, how we transition it to operations, and how we support it day-to-day.

Internet businesses and OTT players continue to take market share from incumbent players, such as telecommunications operators and Digital Service Providers (DSPs), who are slow to adapt to this new reality. This represents a clear threat to DSP revenues.

The digital enterprise is the new normal. The market demands the advantages of the digital experience:

ROADS: Real-time, On-demand, All Online, Do-It-Yourself (DIY), Social.

Most importantly, this experience must not only be ***driven*** by the user, but there must be a **zero-fault customer experience**. On the Internet, there are no maintenance windows, only downtime. Digital means that services must be delivered immediately, not by overnight batch job. It also means the ability to rapidly evolve customer offerings based on real world feedback. That means changes to the way systems are designed and built, how they are operated and maintained, and how they are monitored and how faults are responded to.

All of this requires a new way of architecting, constructing and operating ICT solutions.

2.1 AGILE, LEAN AND THE DEVOPS LED TRANSFORMATION

The new digital approach to product and service development takes much of its heritage from the areas of lean manufacturing and operational management theory. This is about applying industrial design techniques to the software industry.

A key philosophy behind DevOps is the focus on experimentation and quick feedback loops, and the use of minimum viable products that are evolved rapidly based on this feedback. A key tenet of Lean is the relentless drive to simplify and weed out unnecessary work. All of this is intended to maximize customer value. Lean and DevOps put the customer experience at the heart of what they do. They do this through design thinking and breaking down silos to ensure a holistic view of the end-to-end customer experience.

For Internet businesses, everything is about scale. Automation is driven into the development, delivery, and operations lifecycles. For true digital services, this includes autonomous responses to changing conditions, such as auto-scaling in response to demand, and auto-repair of failed components. It also means automation of the entire release pipeline from idea to production. This is DevOps ground zero. Taking ideas such as continuous integration, continuous delivery, and continuous deployment and building a culture around shared ownership and velocity from idea to production.

The result is a focus on tooling, automation, and culture that challenges existing ICT practices.

2.2 A Radically Different ICT

The contrast between traditional "waterfall" oriented ICT and modern "iterative" ICT is stark, and reaches across organizational boundaries:

Old World Approach	Digital World Approach
Culture	
Organizational silos and hand-offs	Shared ownership and high collaboration
Fear of change	Change seen as principal work flow
Manual fulfilment of most functions	Ubiquitous automation
Service Creation	
Segmented products, introduced rarely	Personalized digital life
Software Development	
Waterfall	Continuous and Incremental: Agile, DevOps
Quality Assurance	
Manual testing	Automated build pipelines and quality assurance
Infrastructure	
Application specific, dedicated infrastructure	Open but complex cloud-based infrastructure
Application architectures defined by network design	Network design defined by application architectures
Bespoke infrastructure built once, then maintained	New, cloud-based infrastructure created for each deployment
IT Operations	
Manual configuration changes to critical infrastructure	Automated deployment to all environments
Risk managed through change windows	Risk managed through progressive activation
Pre-defined alarms and manual responses	Autonomous real-time response to error conditions
Processes	
Processes biased towards "build once"	Processes re-engineered for high volume, rapid throughput of changes

In the new digital landscape, services must react rapidly to dynamic conditions, scaling rapidly in response to the unpredictable nature of changing demand or component failure. This ability to scale and respond autonomously is a key driver in the re-engineering of digital applications.

The new digital world has been enabled by foundation technologies of: Cloud and virtualization, APIs, and microservices. These technologies have provided an opportunity to re-examine the way in which ICT is managed; to distil the key processes down to their essence, and re-apply those principles in a way that leverages these technologies to provide a better way to manage risk and change.

This is an important point and worth repeating. The new digital world, when implemented properly, operates not only faster and more efficiently, but with lower risk and increased quality! The 2016 State of DevOps report[1] highlights the value of adopting DevOps practices:

	2015 Survey **Super High vs. Low**	**2014 Survey** **High vs. Low**
Deployment Frequency	30 x	30 x
Deployment Lead Time	200 x	200 x
Mean Time to Recover (MTTR)	168 x	48 x
Change Success Rate	60 x	3 x

This table shows the relative results for key velocity and quality metrics in organizations that have embraced DevOps practices to varying degrees. As practices continue to mature, organizations see their lead time and success rate for changes increase, and their ability to deal with failure conditions improve substantially.

These gains are achieved by looking at the key practices of configuration management, change management, release management, quality assurance, and the software development lifecycle, and re-architecting them to leverage high levels of automation and new ways of managing infrastructure and code release.

The resulting digital platform forms the basis for innovation in a rapidly evolving world. The techniques and principles discussed in this paper are equally relevant in any IT, CT, ICT, CSP, DSP or other setting. i.e. This approach to computing (including networking) will form the basis of the next generation of IT based enterprises.

2.3 A Transformation Challenge without Precedent

When viewed in its largest context, this is a transformation journey without equal and one of the most radical re-inventions that most enterprises will have undergone in at least a decade, and possibly in the company's lifetime.

Such a bold statement requires further explanation.

The traditional enterprise view is to see new product introduction as a project, performed relatively rarely (every few months to a year). Each such introduction project is seen as complex, expensive and risky. It is funded and tracked separately from normal "business as usual" work, follows a progressive "waterfall" process from requirements to design to implementation, and delivery is managed by project managers

[1] https://puppetlabs.com/2015-devops-report

(sometimes several). The entire enterprise (not just IT) is setup around this product introduction philosophy.

By contrast, digital products evolve frequently and rapidly. This *is* the normal business-as-usual process. Products are developed, tested and released incrementally, via agile and DevOps methodologies. New projects start rarely, but evolve continually. Change is the most important work of the company.

By contrast, digital products evolve frequently and rapidly. This is the normal business-as-usual process.

To be successful, a digital business transformation requires re-tooling the entire enterprise and its operating model, from relatively rare, manual activities, to an industrial model where changes are churned out at an increasing rate via a “production line”. Processes must be re-designed to support frequent, hands-free delivery, or they will not scale.

The question is: How do you make fundamental changes to your business processes without putting services at risk? How do you give the patient a heart transplant while they’re going for a run?

3 The Nature of Digital Services

The old way of working was based on expensive, bespoke production infrastructure, and cheaper similar – but not identical – non-production versions. A company may spend $20M on their production environment, but only $20k or $200k on all of their non-production environments combined. This approach was then combined with IT outsourcing to produce a particular view of how infrastructure is built and operated. This, in turn, led to a set of processes and practices that correspond to the risk profiles inherent in this approach to infrastructure.

> *The digital world is built on a few key technology pillars: cloud, APIs, and microservices.*

For example, the expensive nature of the production infrastructure meant that it had to be maintained in-situ as a "snowflake" server (unique, dedicated, bespoke configuration), highly customized to the task at hand. Separation of controls meant that developers did not have access to production. Outsourcing meant that the people implementing the changes had no direct knowledge of the change, its origin, or context, and worked based on "Change Implementation Plans" prepared (usually in MS Word) and then executed manually via cut-and-paste. All of this meant high risk of user error, configuration creep, toxic end state, and other failure modes. The risk management practices, such as change windows, back-out plans, and co-pilots were all developed to deal with the failure modes present in this way of working.

The digital world is built on a few key technology pillars: cloud, APIs and microservices. This combination provides for some very innovative approaches to provisioning and lifecycle management of infrastructure, including:

Infrastructure as code

The ability to create, modify, and delete infrastructure on demand, via API, allows for infrastructure to be created by program. This ensures it is identical each time, created quickly (without human intervention or error), and can be re-created or modified as required, just as easily as re-compiling a program.

"Ephemeral" (short lived) infrastructure

This ability to create infrastructure on demand also leads to an interesting variation, where we can now have short-lived infrastructure. New infrastructure can be created for each new test environment or new product version, and these can be used and then discarded when their job is done.

For production this is especially compelling, offering a way to control the traditional problem of entropy and uniquely configured "snow flake" servers, by ensuring a clean, well-defined environment for each new version.

Progressive activation

One of the major factors in creating legacy change management approaches was that infrastructure was large and bespoke, and had to be changed in-situ. This meant changes had to be completed or rolled back. This "all or nothing" approach made changes expensive, often taking many hours and involving service downtime. This contributed to an organizational fear of change.

Part of continuous delivery and continuous deployment is the ability to dynamically enable or disable code functions via external command. These so-

> called "feature flags" can be used to **decouple deployment from activation**. This, in turn, provides a highly flexible way to control risk associated with code changes and new features.
>
> Rather than risk being controlled based on whether it has successfully been installed or not (an all-or-nothing change), the controlled activation of a deployed feature allows fine-grained control over who sees a new feature when, and also provides for rapid disabling of a misbehaving feature. This means that deployment can proceed frequently, and include partially complete features (no need to carefully integrate only complete features).

The result is that the basic principles underlying good infrastructure management can be reapplied in this new context. Digital operations does this in a way that challenges all assumptions, resulting in operational management practices which are, at first glance, radically different, but which apply the very same principles of good risk management to deliver better results than traditional practices.

3.1 The Evolving Nature of Failure and Availability

In the cloud, the application must deal with failure

The traditional view of infrastructure and failure was to protect the application from as many infrastructure related failures as possible. This reflected a separation of responsibilities between application and infrastructure teams. This protection was usually in the form of redundancy: RAID for storage, redundant network paths, redundant power supplies, active virtual workload migration, and any number of other examples.

Underlying this approach was a school of thought that applications should not need to know about their infrastructure, and an acceptance that failure is expensive to deal with and applications were usually very poor at dealing with most failure scenarios, so should be protected from them. The reality has been that, even when applications do handle failure, it is generally limited, and there is often still a degradation or disruption of service.

This approach meant that up to 50% of the infrastructure was sitting idle, effectively doubling the hardware costs of building and operating each application, with no additional usable capacity. (With a single redundancy, half the infrastructure must be able to support 100% of the load, and so redundancy implies holding half the capacity spare in order to deal with failure; with more redundant components, a single failure takes a lesser percentage of capacity offline, and so less over-provisioning is required.)

The move to cloud computing has seen this view turned on its head.

In the cloud, the application must deal with failure. This decision allows the provider to remove all redundancy from the infrastructure, in turn significantly reducing the cost and complexity of the infrastructure build, allowing them instead to create separate, isolated pools of infrastructure ("availability zones"). Clouds make extensive use of commodity hardware this way in order to pass on the economic benefits to consumers.

This move from ***infrastructure redundancy*** to ***application resiliency*** is at the heart of digital applications. The applications and components of a digital service are smaller, and internally resilient – *to the extent* *they need to be*. Putting the responsibility for resiliency with the service allows it to make decisions about how best to effect the required level of availability and, in turn, resiliency. A component might not have any resiliency, it might have delayed consistency ("eventual consistency") using a NoSQL database, or it might maintain high consistency between different availability zones in order to ensure the best customer experience.

The result of this approach is higher levels of service availability even as the scale and complexity leads to higher numbers of individual component failures.

Netflix is the poster child for this new model. When an entire **Amazon (AWS)** availability zone went down, their service kept running, and customers kept watching movies. In order to achieve this, not only have they architected individual components to span availability zones, but they have created special software, the Chaos Monkey, to actively bring components down in order to ensure their service autonomously responds to the failure, and customer service remains unaffected.

3.2 THE DIGITAL LIFECYCLE – AN ARCHITECTURAL TRANSFORMATION

Digital services put much more power and responsibility with the developer. This has impacts across the lifecycle. The developer now has primary responsibility for testing, packaging, infrastructure definition and build (via cloud management APIs), software release, feature activation, and more.

The nature of the Software Development Lifecycle (SDLC) has evolved to encompass an iterative development and deployment model based on Agile and DevOps practices. This moves away from the more linear waterfall process most enterprises have historically used. This, in turn, requires that: (a) the architecture be more iterative in nature, and (b) that operational aspects of design (previously called "non-functional requirements") are placed at the center of the service design, to meet the new responsibilities for service availability, operability, and scalability previously discussed.

Further, digital services take a smaller form based on microservices, which are easier to design and assure, due to their simpler nature. A larger number of these component services are then combined to form more complex services. This, too, must be reflected in the system and enterprise architectures.

A new way to think about system architecture and enterprise architecture is required; one that is more iterative and incremental; one that embodies microservices and this new holistic, integrated approach to service design.

This related piece of work, to evolve architectural standards to reflect a more iterative, microservice oriented view is also being undertaken by Huawei as part of its contributions to standards bodies such as The Open Group (authors of the TOGAF standard).

4 THE FUTURE MODE OF OPERATIONS

The Future Mode of Operations (FMO) is Huawei's answer to the question raised earlier – how to support both a digital ecosystem and legacy services, and the complex transformation journey between the two, whilst not missing a beat. It is a new operating model for the digital enterprise.

It is important to capture not only the needs of the digital enterprise, but also those of the legacy enterprise, and create a hybrid operating model. Large enterprises have invested tens to hundreds of millions of dollars in their pre-digital world, and any transformation that does not carefully plan for the orderly transition of services is doomed to fail.

> *The Future Mode of Operations (FMO) is Huawei's answer to the question raised earlier – how to support both a digital ecosystem and legacy services, and the complex transformation journey between the two, whilst not missing a beat.*

4.1 WHAT IS AN OPERATING MODEL?

An **Operating Model** is an abstract and ideally visual representation (model) of how an organization delivers value to its customers or beneficiaries.[2] Or to say it another way, people talking about new operating models could mean almost anything! Most discussions around digital transformation take a very narrow view of what the operating model entails, looking only at select practices, such as software development practices. Such a restrictive view will at best limit, and at worse, cause a failure to deliver the benefits of digital transformation:

> *In a Boston Consulting Group analysis, two-thirds of companies reported receiving 50% or less of expected benefits. As well, on average only 33% of significant IT projects have been fully successful since the year 2000. Worse, that figure falls below 20% for projects larger than US$3 million…. **Our own recent research indicates that in excess of 50% of operational executives see at best "some" benefits from digital technologies implemented so far**. This indicates at least a gap between mid-level management teams' perspective and C-suite's ambitions which has the potential to derail change. [4]*

One useful definition refers to five key elements in an operating model:

1. Process
2. Information Systems
3. Locations and buildings
4. Organization and people
5. Suppliers and business partners

[2] https://en.wikipedia.org/wiki/Operating_model

The previous chapters have laid out the all-encompassing nature of a true digital change; not only the way digital services are deployed and managed, but how ICT operations are performed, the way assurance is done, and the very nature of product development.

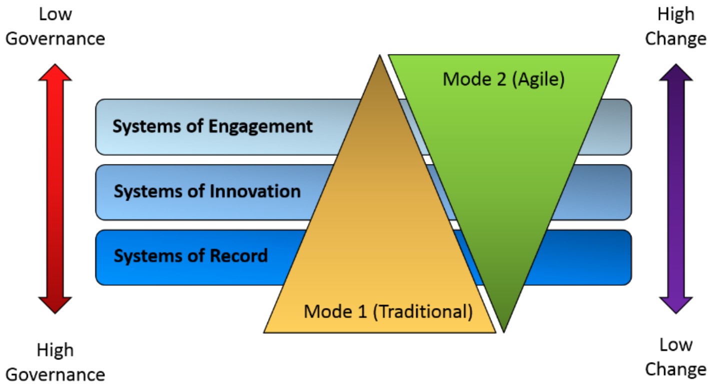
The transformation must be ICT led, and requires significant changes to processes, organizational structures, and KPIs, just to name a few areas. It also means reevaluating existing supplier relationships.

A true digital ICT transformation, however, must extend beyond the walls of ICT and into the business lines. An outside-in view relies on a closed feedback loop from the customer, through marketing and customer support, to the product development team. DevOps supports decentralizing development into product teams that might sit outside the central IT organization. No one structure is correct for everyone, and no structure will be static. Part of the power of the Lean model is the continual and collaborative improvement to tools, processes and practices.

4.2 A Hybrid Operating Model

The digital ecosystem requires a new operating model – a rich, **multi-modal**, multi-speed environment that supports agility without compromising stability. Simplistic early attempts to embrace digital services have seen less evolved thinking around operating models, particularly the trend of Bimodal IT. This should be deprecated, and a real discussion of transformation undertaken.

> *The digital ecosystem requires a new operating model – a rich, multi-modal, multi-speed environment that supports agility without compromising stability.*



The above diagram is a common representation of Bimodal IT, as proposed by Gartner. Bimodal advocates leaving the old systems, governing processes, and teams alone (mode 1 organization), and creating a new digital (mode 2) organization. **This is counter-productive on many levels**:

1. The digital ecosystem does not exist in isolation; it must interact with the legacy systems of record. The nature of this interaction can become a source of friction if the legacy systems and governing processes remain unaltered;
2. It postulates that there is a clear separation between systems that should be considered digital and those that should not. This boundary is arbitrary and limits the value of modern digital

approaches described in the previous section to only a small subset of systems, somehow implying that these techniques are not appropriate to legacy systems. This, in turn, limits the competitive advantage of any ICT transformation and can significantly slow introduction of more significant changes;

3. It tells the incumbent organization that they have nothing to offer the digital ecosystem, and sets up an organizational conflict. Worse, it tells them they must continue to manage legacy whilst a new team manages digital – effectively limiting their career prospects.

Industry analysts are beginning to articulate these and other flaws in the Bimodal model:

> ***Bimodal IT Will Not Work In the Age of the Customer***
>
> *Bimodal IT is about incremental change to the status quo. It does not talk about the broader business, process, and organizational changes required to improve customer experience and operational excellence.*
>
> ***Bimodal IT Has Fatal Flaws over the Long Term***
>
> *When the business needs simplicity-driven innovation, bimodal IT adds complexity with its siloed, two-class system around a fast and slow approach. It makes no sense to have two groups competing for funding, resources, skills, and the business' attention.*
>
> ***A Bolder Business Technology Strategy Will Close the Speed Gaps***
>
> *CIOs need a BT strategy that looks across the business and tech management and holistically makes the changes required to drive simplicity, embed design thinking with Agile, help the business gain program management expertise, and implement more modern application and cloud architectures*
>
> *- Forrester Research [6]*

Of much greater value is to consider the landscape for a true transformation to an all-digital world. This must encompass naturally digital service offerings, but also how to migrate legacy services towards digital replacements. It must support multiple paths through key governance processes, of varying weight, in order to provide appropriate balance between risk and velocity.

This allows for a roadmap to be created, providing certainty around investment, value, and timeframes for such migrations, and how to embrace digital approaches in traditional ICT design and operations.

A true hybrid operating model has the following characteristics:

- Supports multiple speeds through the various governance processes (e.g. change management), based upon objective criteria (risk, activation controls, nature of change, etc.).
- Adopts an "API first" culture to drive innovation into existing systems of record, and enable ubiquitous automation across the application portfolio.

- Sees all systems as capable of benefiting from digital practices. Reviews and tunes operational practices for each application accordingly (versus static operational procedures agreed at time of operational acceptance).
- Seeks to continually improve existing operational practices and tooling to remove friction and further promote an all-digital future.

To achieve a multi-modal organization such as this, core systems (e.g. change management, configuration management, alarm and assurance) must be upgraded to support multiple alternate work flows, with varying degrees of control and self-service, and APIs to drive automation of workflows.

4.3 Shadow IT and Multi Speed IT

A major factor in proposing Bimodal IT as a solution for introducing digital practices into legacy organizations is a frustration with traditional IT, which is seen as having a "one size fits all" attitude to IT. Typical implementation of management processes, generally based on the ITIL service management framework, have created highly manual, people focused processes around project delivery on the one end of the value stream, and production release on the other.

Typical company budgeting cycles exacerbate this problem, making it difficult (in both lead time and business justification) to secure the funding (capex) to initiate the heavy weight of a formal project, and to wait for delivery. On the other side, IT operations typically treat all changes as equally risky, and so reduce progress to the lowest common denominator. Changes to legacy systems can require up to 18 months lead time to be scheduled into the "enterprise release" cycle.

To get around these impediments to progress and deliver value to the business, one of two things happens – "shadow IT" or Bimodal IT. Shadow IT is where the Line of Business (LOB) uses money it has to circumvent central IT and deliver a solution. The availability of cloud computing has made this very easy – one can spin up instant compute power with just a credit card.

This use of shadow IT can introduce significant risk into the business. At its least harmful, it undermines efforts of enterprise architecture, creating duplication, and possibly distributing data where it is not available for other purposes. More harmful is that LOB teams may not understand the impacts of their decision, potentially opening the enterprise up to significant risk in terms of security, privacy or other legal requirements for data governance. Then there are the issues of reliability. One reason for central IT's slower approach is to ensure all relevant matters have been considered during design and implementation. There can be an order of magnitude difference in effort between building a basic system and building one that is enterprise production ready.

Shadow IT will always exist where central processes are significant impediments to a LOB delivering on its objectives and corporate Key Performance Indicators (KPIs). It is no small irony that embracing the digital ecosystem significantly reduces the need for shadow IT by making the IT organization more agile and efficient, and by supporting distributed development efforts that would normally be considered to be in the shadows.

The more centralized solution to this problem is "Bimodal IT" – to create an entirely separate IT organization that plays by digital rules, leaving the old IT organization to continue with its existing practices. This, including the severe limitations of this approach, has been discussed above.

A better solution is multi-modal IT. This is a reflection of the true multi-speed nature of IT environments. Not all systems are equal. Not all changes have the same risk profile. A comprehensive implementation of a digital model includes having many speeds through “the system”. (Not two different systems!) This is a direct result of the “eliminate waste”, simplify, and automate mantras of applying Lean methodologies.

Production Change Management (what IT Operations does) is a prime example. The purpose of any production change management process, whether based on ITIL or not, is 2-fold: (1) to qualify the correctness of the new (previously unseen) target state, and (2) to qualify the process by which you transition to that new state. When seen this way, it is clear that for truly new changes (the default setting), one must learn about and validate the new target state and the change process. Traditionally, due the relative cost of infrastructure, this was done through having a variety of environments, progressively closer to production in terms of configuration and behavior. This is, in effect, a “controlled learning exercise” – an attempt to avoid unforeseen issues when applying the change to production.

Once a change has been seen before (e.g. adding a new disk partition), that **change can be standardized via procedure or automation**, and takes on a new, lower risk profile. High volume changes (such as adding a user) will become “standard” (risk 0) changes, and not need any change management governance at all.

It should be clear that, for any application, one might draw up a number of classes of change, each with different risk profiles and governance procedures, rather than applying a “one size fits all” model of governance artefacts, change approvals and scheduling.

This approach of optimization is at the heart of multi-modal IT and embracing a DevOps culture for Continuous Integration, Continuous Delivery, and Continuous Deployment. Engineering effort is invested to streamline the most important (e.g. most frequently executed, highest risk, more error prone) changes and to ensure they are low risk and invisible to the customer base.

4.4 Why This is So Important for the Telecommunications Industry

Telecommunications operators have generally invested hundreds of millions of dollars each in their infrastructure and supporting computing systems. They cannot afford to simply replace or sideline these systems as they transform to support a digital mode of business. They must adapt to this world, and do so rapidly.

The nature of this investment requires that it must be both protected and opened up to participate in the new wave of digital services. This means transforming OSS and BSS systems. It also means taking hitherto closed resource pools, such as traditional network elements, placing them behind APIs, and advertising them via technical service catalogues, so that applications (and potentially customers) can self-service infrastructure changes.

As detailed elsewhere, it also requires significant changes to established operating practices and tools, and instituting a new culture.

All of this is necessary in order to meet customer expectations for how services are delivered in a digital world.

4.5 What is the FMO?

The FMO is a target state, a fully functional hybrid operating model. It is also a transformation roadmap; an examination of how to evaluate current state against this proposed target (maturity level assessments, benchmarking, etc.), and the strategies and tactics to give effect to the transformation.

What is FMO? It is a discussion. It is intended to raise the issues that affect Huawei and its customers, and to help drive the conversation towards better software and consulting products, and better customer relationships.

What might the outcomes of this discussion include?

- Architectural frameworks and addendums to standards such as TOGAF and IT4IT;
- Guidelines for software development teams;
- Improved product management guidelines;
- Maturity Level Assessments for CT and DSP customers;
- Transformation guidelines and case studies;
- Training for Huawei staff and for customers.

This is the first of a series of white papers discussing the FMO. Future papers will examine:

- The transformation journey in detail;
- The future of consulting services in a digital world;
- The impact of digital applications on business and enterprise architecture.

4.5.1 The FMO as Architectural Guidelines

The FMO represents a new way of designing and evolving systems, a much more holistic one. Traditional architectural approaches and frameworks embody a "design then build" philosophy, requiring that an architecture be completed before moving onto coding, and then testing, delivery, etc. Just as the SDLC adapted to agile software development with new frameworks, such as SAFe[3], the approach to architecture must similarly evolve to take a more incremental view. (Agile expresses this as "design for today, not for tomorrow".)

The expanded set of application responsibilities, in areas such as reliability, scalability, upgradeability, and enhanced production controls such as feature flags, all need to be reflected in a newer approach to architecture. This is not simply making architecture iterative, but adding significant new areas of concern to frameworks to ensure the resulting services meet modern cloud native criteria.

The FMO is a natural lead-in to such an addendum to existing frameworks and standards for the digital era.

4.5.2 The FMO as Product Guidelines

One aspect of the FMO is to provide guidelines for delivering better software. Just as it must lead to changes to architectural practices, it must also lead to design and coding practices. These "cloud native

[3] Scaled Agile Framework (SAFe): http://scaledagileframework.com/

product guidelines” should be defined and published, and then used across the company and in the wider community.

This will serve software teams within Huawei in their quest to be world class providers of software, and to ensure that software delivered is aligned with, and promotes, digital services via the FMO. It is important that Huawei deliver software that is cloud native, API based, and provides a best of breed digital experience. To that end, a checklist of requirements can be created from the FMO which will help ensure software is compatible with the goals described in this white paper.

> *These “cloud native product guidelines” should be defined and published, and then used across the company and in the wider community.*

For example, software vendors should stop writing custom installation scripts. These are a great source of angst for system administrators and rarely succeed without significant additional work. They impede clean upgrades and hinder platform patching and upgrade cycles. Better would be to provide configuration files for software configuration management tools, such as Docker, Puppet, Chef, or Ansible. The practices around installation and upgrade of software would form one section of the guidelines.

Huawei could also use these guidelines to demonstrate thought leadership. We could, for example, publish these and nurture their development by the wider community under an open source license. This would provide a way to capture and promote best practice, and shape community attitudes.

4.5.3 The FMO Guiding Product Management

Huawei’s products must evolve in their design to be cloud and digital natives. This has implications on the way that product management is done, on product roadmaps, and on feature prioritization. Present product groups take an inconsistent view on these matters, and the FMO can help drive a singular focus and consistency across the product portfolio.

This would drive consistency across API catalogues, microservices, and orchestration, which would unify and amplify the product line offering.

4.5.4 The FMO as a Service Portfolio

Having established the value and transformation story, a major outcome of FMO should be demand for services assisting with the ICT transformation described by the FMO.

Traditional IT outsourcing won’t exist in its present form for long. It was based on a broken model that was perpetuated for two decades. The bespoke nature of IT environments meant that a ***process*** view of outsourcing (Business Process Outsourcing) was never appropriate, and that a ***project*** view was required[4].

IT outsourcing, however, was designed based on a process model. This caused significant conflicts where, for example, continuing pressure to reduce costs could not be met with automation or other techniques due to the bespoke nature of the systems. Attempts to drive consistency across the environment were met with resistance, and only small improvements were ever achieved.

This tension always saw ITO providers presented in a negative light and made it difficult to establish a peer or key partner relationship.

[4] A **project** is for work that is always unique, whereas a **process** is for work that is always the same.

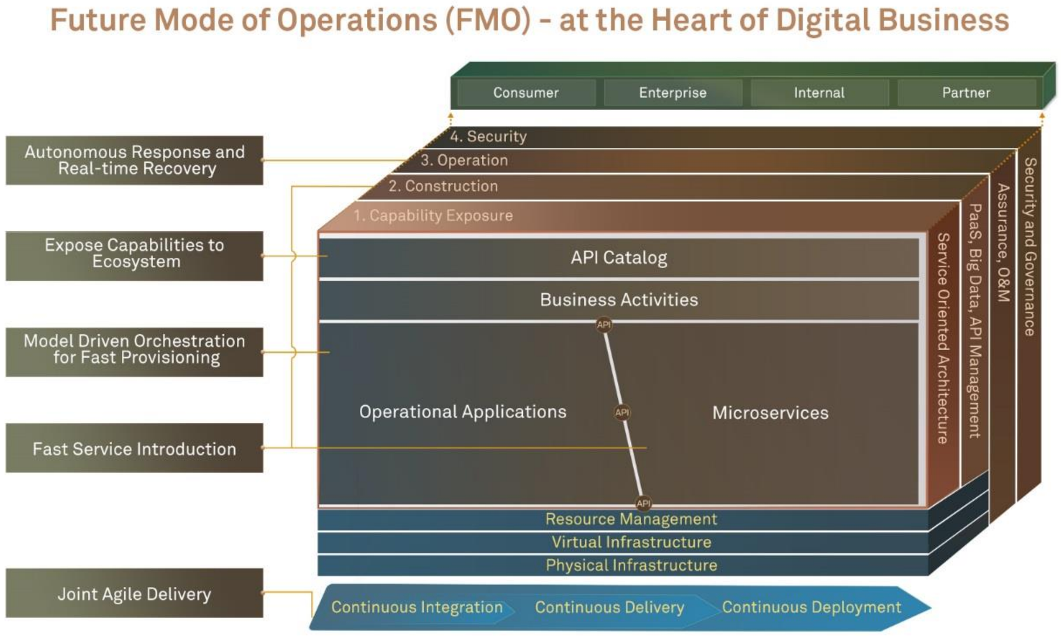
The nature of this transformation is that it can provide an inflection point for Huawei to become a trusted partner on ICT transformation. As the world transitions to a cloud model, the nature of infrastructure has changed forever. Bespoke hardware and configurations give way to automation and "infrastructure as code". Similar infrastructure pools (e.g. compute) are all placed behind the same APIs, making operations more efficient, and applications more portable than ever before. Moreover, this makes it possible for vendors be benchmarked against industry-wide metrics.

Huawei is well placed to assist, not just with incremental product upgrades, but with major transformation initiatives such as FMO. To do this, however, will require investment in a new, higher value consulting practice that is more business and organization focused than traditional technical outsourcing. Services here might include:

- Maturity Level Assessments (MLAs) and Performance Benchmarking;
- Transformation Consulting Services;
- Prime System Integration on transformation work.

Work would include creating the checklists behind the MLAs, performing pilot projects and developing case studies, developing top-down and bottom-up roadmaps for transformation, and educating customers on the changes this new mode of operations requires.

5 The FMO Cube



The FMO cube provides a holistic view of the different aspects of building and running a digital enterprise. Each slice highlights one key aspect of service delivery and operations.

5.1 The Capability Exposure Slice

The **Capability Exposure** slice shows the business capability perspective: how increasingly complex applications are developed atop the layers beneath, from raw physical and virtual infrastructure, to cross-domain infrastructure services, to application components and microservices, all ultimately exposed in the form of an API service catalogue. It is these capabilities that form the applications of the digital enterprise, such as portals and other engagement channels, and innovation applications.

This is a relatively standard architectural view, but where these components might normally not be visible outside the architectural diagrams of a traditional monolithic application, in the digital world these components all exist independently, and advertise and consume other services via API.

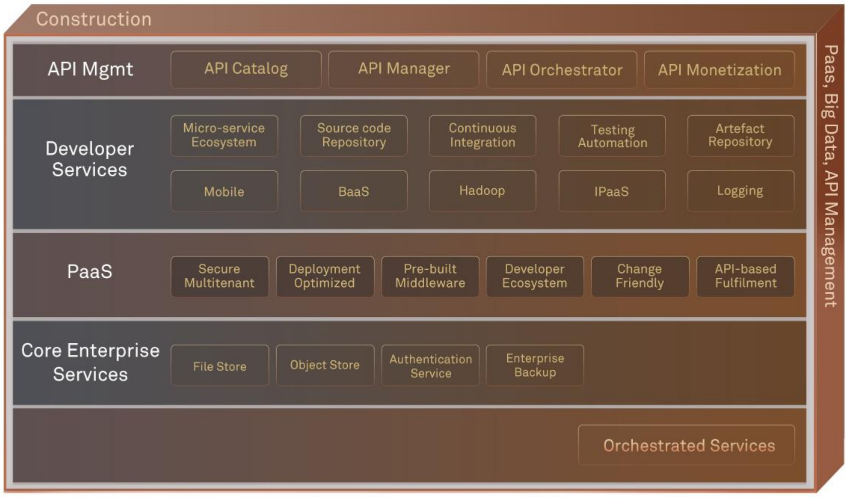
Participating in the wider developer ecosystem allows the DSP to evolve services at the speed of the market, and not at traditional enterprise speed.

This approach creates a rich ecosystem where both digital microservices and legacy systems of record interact to provide rapid delivery of new services.

Ultimately, these are exposed via APIs for wider consumption: by internal applications (such as portals and other customer engagement channels), by enterprise customers, by partners, and by the wider developer and consumer community.

Part of the value of the digital enterprise is that the Digital Service Provider (DSP) is no longer responsible for developing the entire product stack from top to bottom, but can also participate, both as a consumer and as a provider of component services, in a rich developer and service ecosystem. This is important. It allows the DSP to evolve at the speed of the market, and not at traditional enterprise speed.

5.2 The Construction Slice



The **Construction** slice provides the software development view, from build environments based on containers and PaaS, through the tool chain of the release pipeline (including CI/CD tools, source and binary artefact repositories, testing automation, and configuration management), to developer-assisting services, such as Backend-as-a-Service (BaaS), and central IT functions (such as storage), to the exposing of new services via an API management layer. The emphasis of the construction slice is to make it as easy as possible for developers to be productive.

5.2.1 Software Development and Continuous Delivery

At the core of the digital enterprise is a new, highly integrated approach to the software development lifecycle.

Where once developers, quality assurance, and operations were all separate silos, and code was formally handed off between them, the digital world embraces ***continuous integration***, ***continuous delivery***, and ***continuous deployment***. At the core of all of these is the definition and automation of an end-to-end release pipeline – from idea to production.

This pipeline uses a number of tools to assist with automation, including: source code repositories (such as Git, and their user interfaces, such as Github); binary artefact repositories (e.g. Sonatype Nexus); Continuous Integration (e.g. Jenkins), build automation (e.g. Rake, Maven, Ant), test capture and automation (e.g. Cucumber, SOAP-UI, and many more); deployment automation (e.g. Puppet, Chef, Ansible), infrastructure construction (API-based Cloud orchestration), and more.

In a DevOps world, all members of the cross-fertilized team work on continually improving the build pipeline tooling. The result is that as the system grows, it becomes more stable, more mature, and more efficient.

5.2.2 APIs and the API Gateway

Traditional IT involved a lot of bespoke application-to-application interfaces, negotiated and constructed over weeks or months. This made applications highly dependent on each other, and added significant delays to projects. The intention behind the Enterprise Service Bus (ESB) was to build more standardized interfaces, but these often still required a series of negotiations to on-board transactional interfaces between applications. The result was a slow, highly centralized view of application integration.

In an API-based services world, the supplying system defines a public interface in the form of one or more APIs. These are advertised in an API gateway along with documentation on how the API is to be used. The intent is to create general purpose APIs that encourage consumption, rather than bespoke interfaces for individual applications.

A consuming application views the details in the service catalog of the API gateway, and subscribes, immediately gaining access to the API and making calls. This process from searching to first consumption takes minutes, and no negotiation or manual handling is involved. This is the power of API based services.

The result is a rich ecosystem of services which consume each other. Applications are assembled from components, like pieces of a puzzle. Developers are empowered to leverage services already written by others, reducing duplication, risk and the all-important time to value. Individual services are more specialized, performing only a single function, or group of related functions; this makes them smaller and more robust, and allows for the rapid evolution of individual services via new APIs and API versions.

5.2.3 iPaaS – Integration (Platform) as a Service

With the proliferation of smaller, single purpose microservices, some local and some third parry, a further issue that must be addressed is that of integration. Each of these services could potentially require or generate slightly different data (e.g. one service wants a parameter of “1”, another wants “64MB”.) Some integrations are more complex than simple data conversion; these might involve “integration flows”.

A useful definition of iPaaS from Gartner is:

> *"Integration Platform as a Service (iPaaS) is a suite of cloud services enabling development, execution and governance of integration flows connecting any combination of on premises and cloud-based processes, services, applications and data within individual or across multiple organizations."*

The use of terminology has caused some confusion. iPaaS and PaaS are not related. iPaaS is so called because it offers a platform upon which you load and manage integrations. There are both on-premise and cloud based iPaaS offerings.

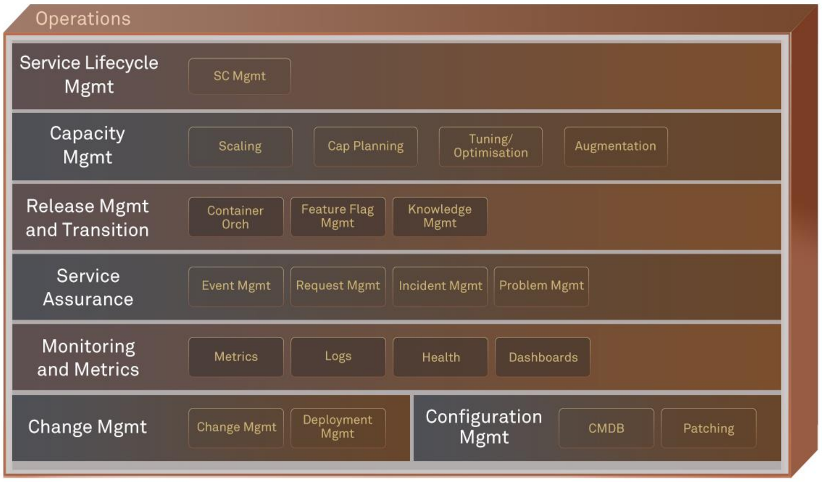
5.2.4 PaaS – Platform as a Service

Another central pillar of service construction is Platform as a Service (PaaS). PaaS serves a number of key functions:

- It provides a consistent "habitat" (settings, conditions, surroundings) across release environments from development through test and staging to production. This consistency is essential to ensuring quality code releases that do not suffer unexpected errors due to configuration diversity between environments;
- It provides tools and APIs to assist with workload migration, orchestration and management. This includes the ability to scale instances quickly;
- It provides a rich (and user extensible) catalog of pre-built images, to aid in consistent, rapid construction of complex applications.

Many people talk about containers and PaaS, but those discussions are focused on software development. For these technologies to succeed, the operational view must also be carefully considered. The third slice, the **Operations** slice, targets the effective operation of digital services.

5.3 THE OPERATIONS SLICE



The changes to Operations practices in a digital enterprise are as significant as those undertaken by software development teams, and a complete Digital Lifecycle Management view is required, if the benefits of a digital ICT transformation are to be realized.

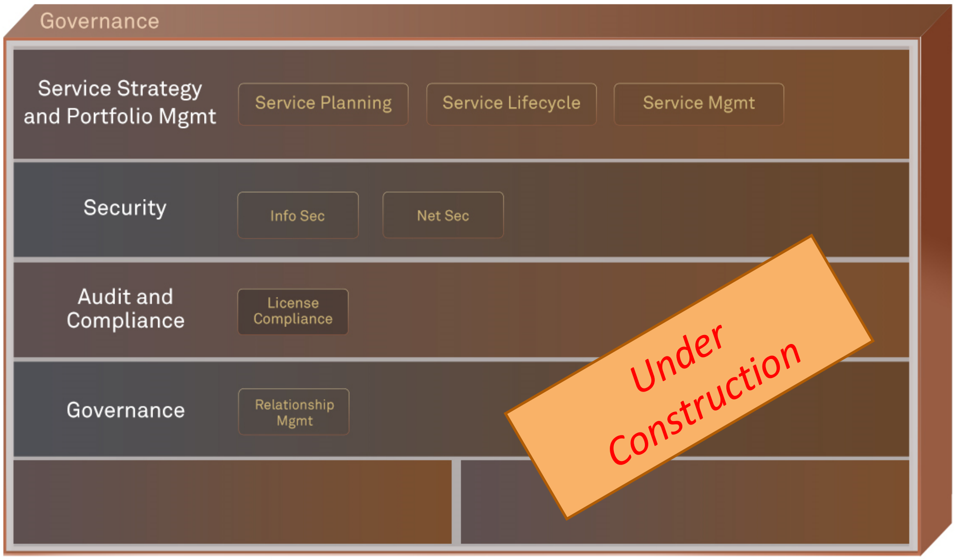
The traditional approach to operations, as embodied by most ITIL implementations, is one that is heavily biased towards manual maintenance of highly customized servers. This makes each change more intricate and error prone, more time consuming, and inherently more risky. As mentioned previously, the operational processes and practices of the IT department have evolved based around this risk model.

Traditional manual processes cannot keep up with digital volumes; the digital enterprise embraces change, and all processes must be re-engineered to support a high volume, low touch approach to change.

Change Management and Configuration Management practices must reflect automated, frequent deployments. Alarming infrastructure for pre-defined failure conditions is no longer adequate (it never really was). Modern systems require autonomous response to any service affecting failures. This is the only way to handle the scale of modern systems.

The modern digital enterprise is not static; it must continue to evolve and optimize its processes and practices to ensure the greatest efficiency and highest rate of change without compromising quality or customer experience.

5.4 THE GOVERNANCE SLICE



The final slice of the FMO cube is the governance slice. As with the Operations slice, many of the functions in this slice will be familiar, and have their counterparts in ITIL, eTOM and similar ITSM frameworks. And, as with the operations processes, the governance processes must evolve to reflect a much more dynamic, iterative model of service definition and delivery.

6 Building the Digital Enterprise

As stated up front, the FMO is not just about operating a separate digital enterprise, but about a hybrid operating model that integrates legacy and digital worlds.

This section looks at various aspects of managing a hybrid environment.

6.1 Operating a Digital Microservice Ecosystem

There are significant challenges in operating a microservice-based ecosystem. The number of components can balloon quickly, and the cyclic dependencies between services can make diagnosis more difficult. Container and ephemeral infrastructure can mean that services are highly dynamic in terms of the infrastructure they reside on, potentially changing second by second. This can make event correlation and problem diagnosis much more complex.

All of this requires significant changes to operational practices and tools. The siloed application perspective of legacy development (Application Development and Maintenance [AD&M]) and production support (Production Support and Management [PS&M]) cannot survive in this new paradigm.

The way in which applications and services are monitored and assured will undergo significant change. There is growing pressure to evolve operational systems to be far more automated, deal with vastly more operational telemetry, and provide better correlation and insight.

Teams must also evolve. The traditional separation of development and operations teams will dissolve, as it has in DevOps teams. KPIs must be shared across the team, rather than the conflicting KPIs that presently impede cooperation. Developers must support the systems they build (and feel the pain of poor design decisions), and operations must evolve into Site Reliability Engineering.

Processes must evolve, too. Release will no longer imply a formal handover between teams, and will become simply another step in the release pipeline. Change management practices will evolve significantly to embrace a nuanced view of change, where objective risk profiles will define the process and quality gates enforced by the pipeline.

6.2 Site Reliability Engineering

Site Reliability Engineering (SRE) is a relatively new role, having been pioneered by Google, Amazon, Netflix, and the other Silicon Valley digital service companies as their solution to how to perform Internet scale operations.

> *SRE is “what happens when you ask a software engineer to design an operations function.”*

As described in previous sections, manual practices cannot scale to meet the demands of digital services and Internet based growth. New tools, practices and processes are required.

Ben Traynor of Google describes SRE as, “*fundamentally, it’s what happens when you ask a software engineer to design an operations function*.” It is about driving automation and autonomous intelligence into operational practices, in order to deal with the scale, dynamic nature, and inherent failure rates of underlying components as services evolve and expand rapidly in response to Internet scale demand.

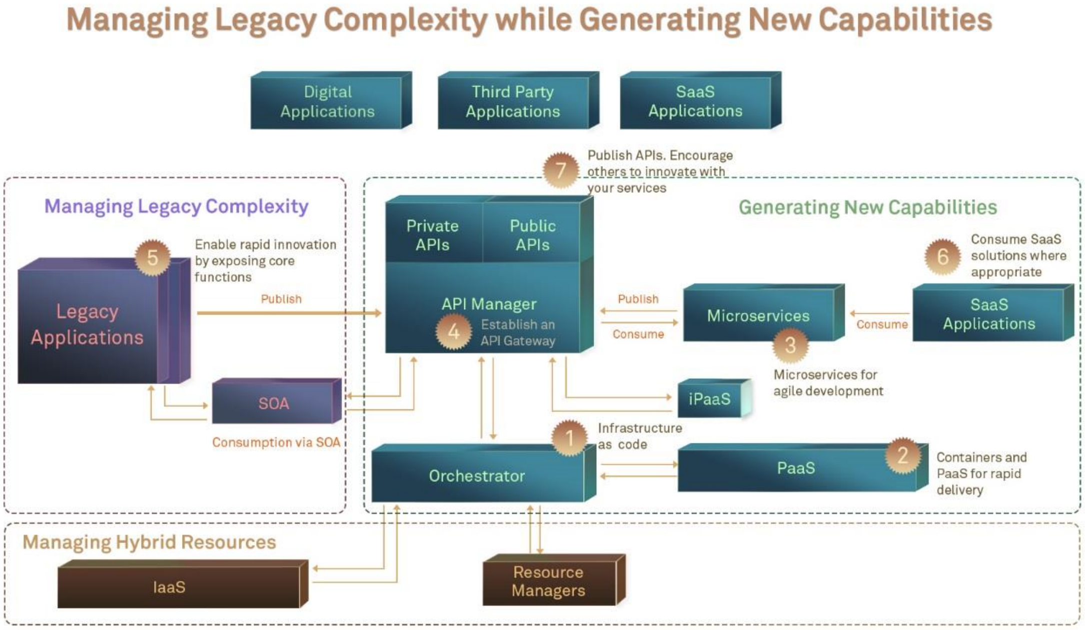
From a service availability perspective, SRE has principal responsibility for all aspects of service assurance, including: operational telemetry, service monitoring and dashboards, service continuity, incident response and root cause analysis.

Looking beyond assurance, SRE has a much deeper role in service quality: latency, efficiency, scalability, performance, change management, capacity planning, and any and all systemic improvements. This responsibility isn't just to determine issues, but to perform the work to address them. SREs have a software development background, and typically spend up to 50% of their time working on these improvement works.

SREs work closely with the development teams as mentors and partners to continue to enhance the state of practice across the organization. This includes guidance on automation and monitoring, ensuring quality of all production handovers, and enforcing a long-term view from development teams.

Part of the SRE mentality is to ensure the system can respond to failures, by exercising those failure modes. This may be manual (operations drills), or the automated introduction of failure into production systems (e.g. Netflix chaos monkey).

6.3 INTEGRATING LEGACY AND DIGITAL SERVICES


Managing Legacy Complexity while Generating New Capabilities
Digital Applications
Third Party Applications
SaaS Applications
7 Publish APIs. Encourage others to innovate with your services
Managing Legacy Complexity
Generating New Capabilities
5 Enable rapid innovation by exposing core functions
Legacy Applications
Publish
Private APIs
Public APIs
API Manager
4 Establish an API Gateway
Publish
Consume
Microservices
3 Microservices for agile development
Consume
6 Consume SaaS solutions where appropriate
SaaS Applications
SOA
Consumption via SOA
iPaaS
1 Infrastructure as code
Orchestrator
PaaS
2 Containers and PaaS for rapid delivery
Managing Hybrid Resources
IaaS
Resource Managers

The Digital Enterprise does not exist in isolation. Larger DSPs have invested many hundreds of millions of dollars over years and decades building complex applications. These run the enterprise, and cannot be easily discarded. Any transformation journey must integrate these two worlds if it is to succeed.

The approach described in this section ensures the safety of legacy systems of record, which continue to operate under their existing tight governance controls, whilst providing a path for rapid innovation, differentiation, and engagement.

This diagram shows the relationship between the key components in the environment, and how legacy might be bridged with the digital ecosystem:

1) New services use Cloud-based infrastructure to create and modify infrastructure on-demand ("infrastructure as code"), without the normal lead times or capacity commitments of the traditional approach to infrastructure ordering and fulfilment;
2) These new services are built on PaaS technology such as containers, providing better isolation from the surrounding infrastructure, and an ability to rapidly scale to meet demand;
3) New applications and services are developed in the "microservice" model – to do one thing and do it well. This ensures a significantly different risk profile, enabling more rapid service evolution;
4) Each self-contained microservice advertises its wares via an API service catalog, managed by an API Management platform. This is how digital services consume; if it is in the catalog, it can be purchased (subscribed to) and consumed in a matter of minutes;
5) The digital ecosystem cannot exist in isolation. There are two ways in which the legacy environments are bridged into this ecosystem: (1) via the Enterprise Service Bus (ESB) platform, and (2) directly via APIs:
    a. Building a REST/ESB bridge[5] allows APIs to be built that can be serviced by legacy applications. This significantly reduces the burden of transformation, allowing it to proceed in small steps rather than giant project-based leaps; and
    b. Major applications can be bridged into the API ecosystem by developing APIs around critical functions, so that the core application continues to be controlled by high-governance change management practices, but innovation can happen much more rapidly in microservices that leverage these published APIs. This provides a critical mechanism to migrate functions out of the legacy monolith into microservices, slowly replacing the legacy application;
6) Part of the ethos of the digital ecosystem is that it involves the wider, richer application and service ecosystem of the Internet. It is often much more cost effective to use a SaaS provider, where a suitable one exists, than it is to develop and maintain each and every building block and service yourself. The use of API-based services makes this decision easier, and one that can be made independently for each service;
7) Finally, use of this ecosystem is not just about using the services of others, but of advertising services for others to consume via API. This turns the DSP into a platform that others can innovate on, and means that innovation happens at the speed of the market, rather than the speed of the DSP. Leveraging APIs in this way provides avenues for: brand enhancement as an innovator, new service revenue, service uplift revenue, and compound service revenues.

In the digital world, Integration-Platform-as-a-Service (IPaaS) performs the functions of traditional Enterprise Service Bus (ESB) with respect to integration workflows between services (both internal and third party).

[5] Representational State Transfer (REST) is the lingua franca of the Internet API economy. It allows providers to define their API offerings in a clear, resource-oriented manner using the HTTP verbs of GET, PUT, POST and DELETE. This consistency has helped drive the adoption of APIs as a new integration approach.

6.4 JOINT AGILE DELIVERY

This FMO approach is not just for green-field organizations. It must support continuous integration and **Joint Agile Delivery (JAD)**. Projects can leverage PaaS, Cloud and APIs to accelerate their development and delivery timeframes. They must then deliver the results into a complex, multi-faceted production environment without disruption.

PaaS optimizes the build, test, release and production control functions. This is achieved through the use of a single, consistent build and release pipeline, from idea to production, which ensures quality through both automation (such as build and testing), and through manual gates, such as code reviews and user acceptance. Continuous Integration ensures clean builds, and coupled with Continuous Delivery ensures that the current iteration is always delivered to the Production boundary, ready to be deployed via JAD processes as required.

This approach supports a more nuanced view of change and risk management, recognizing that not all systems can be treated the same. JAD provides for multi-speed IT development, integration, delivery, and deployment, each operating at different speeds, and with potentially different handover and release procedures.

6.4.1 JAD vs. DevOps

JAD and DevOps are related, but different in significant ways.

DevOps provides an end-to-end view of a particular product line, using a build pipeline, ubiquitous automation, continuous integration, continuous delivery, etc. It gives the (cross functional) development team much more control and responsibility for their product, including production operations.

DevOps is concerned with the culture, practices and technologies that drive collaborative innovation across the entire lifecycle.

JAD, by contrast, is about joint development and delivery into existing operational support processes.

JAD is a series of multi-party interlocks to support agile software development and delivery in a complex carrier environment. It does this through the expected liaison at key points along the way, such as around requirements, testing, and acceptance. It leverages CI/CD tools and techniques to ensure robust, repeatable, rapid delivery of each code iteration. It is compatible with agile software delivery methodologies such as the Scaled Agile Framework (SAFe).

7 THE FMO AND TELECOMMUNICATIONS INDUSTRY TRANSFORMATION

With the new world mapped out, the question must turn to how this could be applied in a carrier or DSP setting.

The telecommunications industry has, due to its large investments in network infrastructure and large customer bases, always considered itself special in terms of how its systems are built and managed. Industry groups such as TM Forum have existed to bring some level of consistency to practices across members.

New Internet industry giants, such as Amazon and Google have invested in computing infrastructure at a rate that dwarfs that of the CT industry; their customer bases also far greater than most CT players. The Internet industry have shown that scale requires a different approach to design and operations.

For operators, applications are generally thought of in terms of OSS, BSS, and a few other groupings. These systems and their governance practices place a limit on how quickly a company can evolve to offer digital services. If an operator is to compete with digital offerings, it must first address these points of significant friction:

- Assurance and O&M. The traditional approach to alarm definition (and response definition) must evolve to include dynamic alarm management via API and autonomous response to events, such as auto-scaling and component reconfiguration or reconstruction. Recovery leads repair, and orchestration provides more advanced methods of recovery such as rerouting and constructing replacement virtual infrastructure, all of which delays or eliminates traditional truck rolls or manual responses;
- OSS. Beyond assurance, OSS includes functions such as inventory and configuration management (CMDB and more), service provisioning and reconfiguration (orchestration), and fault management (event definition and response). The emerging nature of “everything as a service” (XaaS) requires a new life-cycle oriented view to how services are defined, orchestrated together into products, advertised (via API catalog) and consumed (via self-service). There is significant pressure on OSS to evolve into a platform that supports compartmentalized tenancies that evolve much more rapidly, under self-control (of the product team and not IT);
- BSS. The traditional BSS areas of product, customer, revenue and order management, as for their OSS counterparts, must evolve to support innovation and more of a platform design perspective. Product development must be decentralized, to the point that customers can design and tailor their own products using the building blocks provided;
- Platform evolution. The ability to respond to market dynamics, to create new products and services, and apply new business rules all require a significant level of flexibility in the underlying platform, so that it can evolve continually. New product and platform features should be rolled out progressively and frequently via CI/CD/JAD best practices.

In a digital world, all of the above services must be decomposed from slow moving monoliths to agile, rapidly evolving microservice-based platforms.

A key area of focus for companies such as Huawei is how to evolve their product lines toward this converged future. New versions of OSS and BSS systems must be cloud native and encompass new technologies such as SDN and NFV using a plug-in approach that is more extensible and configurable. New technologies will be introduced rapidly, and a generic ability to onboard new resource pools and functions is essential for customer competitiveness.

IES and BES are prime examples of forward looking product thinking. Huawei has recognized the value inherent in this digital approach, and the architectures of IES and BES are evolving to meet the needs of the digital ecosystem, by decomposing each platform into foundation services, and exposing these via API to enable rapid innovation. More work is to be done, and the FMO can help guide the thinking in this domain.

8 CONCLUSION

The future is now. Digital Service Providers have already commenced their journey towards digital service delivery and are already grappling with the changes that this must bring to their operating models. As this paper illustrates, the concepts and practices are straight forward to comprehend and communicate. Huawei has begun moving key product lines to be FMO friendly, and the others will follow. We must build momentum and capabilities around this transformation challenge.

In following discussion papers, I will examine several key topics in more depth:

- The evolution of IT services and products, and what this means for Huawei's business;
- Strategies and tactics that can assist in planning and transforming an organization to support a digital ecosystem, including realignment of functions and organizational structures;
- The effect of digital services on traditional roles and frameworks from architecture to the development and deployment;
- Guidelines for developing cloud native software;
- How FMO relates to key Huawei product lines, such as TelcoOS, OWS, IES, and BES.

9 REFERENCES AND OTHER INFORMATION

9.1 REFERENCES

[1] Eric Xu’s keynote speech, Global Services Forum, September 2015

http://3ms.huawei.com/mm/docMaintain/mmMaintain.do?method=showMMDetail&f_id=SPS15092554010109

[2] Heavy Reading white paper: Aligning Business & Operational Transformation: Enabling the Network to Participate in the API Economy.

http://img.lightreading.com/downloads/Aligning-Business-Operational-Transformation-Enabling-the-Network-to-Participate-in-the-API-Economy-by-Heavy-Reading.pdf?p_redirone=yes&piddl_promo

[3] The Open Group: A Framework for Digital Customer Experience

www.opengroup.org/bookstore/catalog/w165.htm

[4] Putting Digital to Work the Lean Digital Way

http://www.genpact.com/docs/default-source/resource-/putting-digital-to-work-the-lean-digital-way

[5] Harvard Business Review: Lean Strategy. David Collis.

[6] Forrester Research: The False Promise of Bimodal IT. John C. McCarthy and Sharyn Leaver. (April 2016)

9.2 GLOSSARY

CT	Communications Technology
IT	Information Technology
ICT	Information and Communications Technology (the combined world view)
TOGAF	The Open Group Architectural Framework
IT4IT	The Open Group process framework for managing IT environments
ITIL	IT Information Library. A global process framework for managing IT environments
AD&M	Application Development and Maintenance
PS&M	Production Support and Management
SRE	Site Reliability Engineering

9.3 AUTHORS

Principal Author: Geoff Halprin

Geoff has over 30 years of experience in the IT and ICT industries. He has held positions including CIO, VP Operations, VP Engineering, VP Consulting, Director of Network Operations, Principal Consultant, developer, and system administrator. He has spoken at over 20 conferences around the globe on subjects pertaining to IT operations, security and DevOps, and was president of the System Administrators Guide (SAGE), and a board member of the USENIX Association. Most recently he spent 5 years driving the automation and DevOps agenda in a large Telco environment.