Deloitte.

DevOps from an auditor's perspective
A meaningful partnership to drive transformation

27 June 2019

Transformation enabled by DevOps

Challenges with Historic Hosting and Operations Approaches

Benefits of DevOps and Cloud Hosting Approach

Quality

Challenges:

1. **Inefficient deployment** of solutions, maintenance and support

2. **Inconsistent tools, quality and functionality** provided to our people and clients across the globe

3. **Some businesses lack technology resources** to support certain tools locally

4. **Increased demand on internal controls and tool inspections** being driven by policy and regulatory standards

Benefits:

1. **Quicker deployment** of solutions, maintenance & support

2. **Consistent tools, higher quality and functionality** provided to our people and clients

3. **Platform/Software as a Service** provide "Best in Class" hosting

4. **Compliance and alignment** to operational requirements may be more closely managed

Financial

Challenges:

1. Distributed on-premises hosting results in **significant annual expense** (IT personnel, infrastructure, deployment, network, software, support)

2. **Innovative capabilities of competitors** continue to emerge; meeting this challenge is complex with legacy hosting & operations approaches

Benefits:

1. **Lower costs** by rationalizing IT and support infrastructure

2. **Defend / win customers** by developing eminence in the use of innovative solutions and faster time to market.

Security / Technology

Challenges:

1. Challenging and constraining when implementing innovation tools using analytics, big data, etc. where local hosting is used

2. Difficulty in meeting increased security concerns from clients and regulators

3. **Inconsistent security**; greater potential for security breach

Benefits:

1. **Provide scale** and power for cognitive technologies, analytics, big data, etc.

2. Provides a **consistent high-quality message on security** to clients and regulators

3. **Reduced breach exposure** achieved by operating constrained points-of-presence with leading security principles

Reference Cloud Operating Model

Framework Areas of Responsibility

Cloud Operating Model
Adopt an agile operating model to transform into a single workgroup and to foster a culture for global delivery

DevOps Enablement
The foundational operating concept to build and deliver application with minimal manual intervention with a continuous delivery pipeline

1. Cloud Architecture
End to End Platform Architecture that ensures overall design and delivery of cloud services for the platform and automation

2. DevOps Support & Service Management
Developer and operations services to accelerate application creation and standardize. Provide platform support, 1st – 3rd level support and supports Continuous Integration / Continuous Delivery process

3. Security Architecture
A unified approach to cloud security. Embedded into the overall architecture to ensure security standards, compliance and controls exit
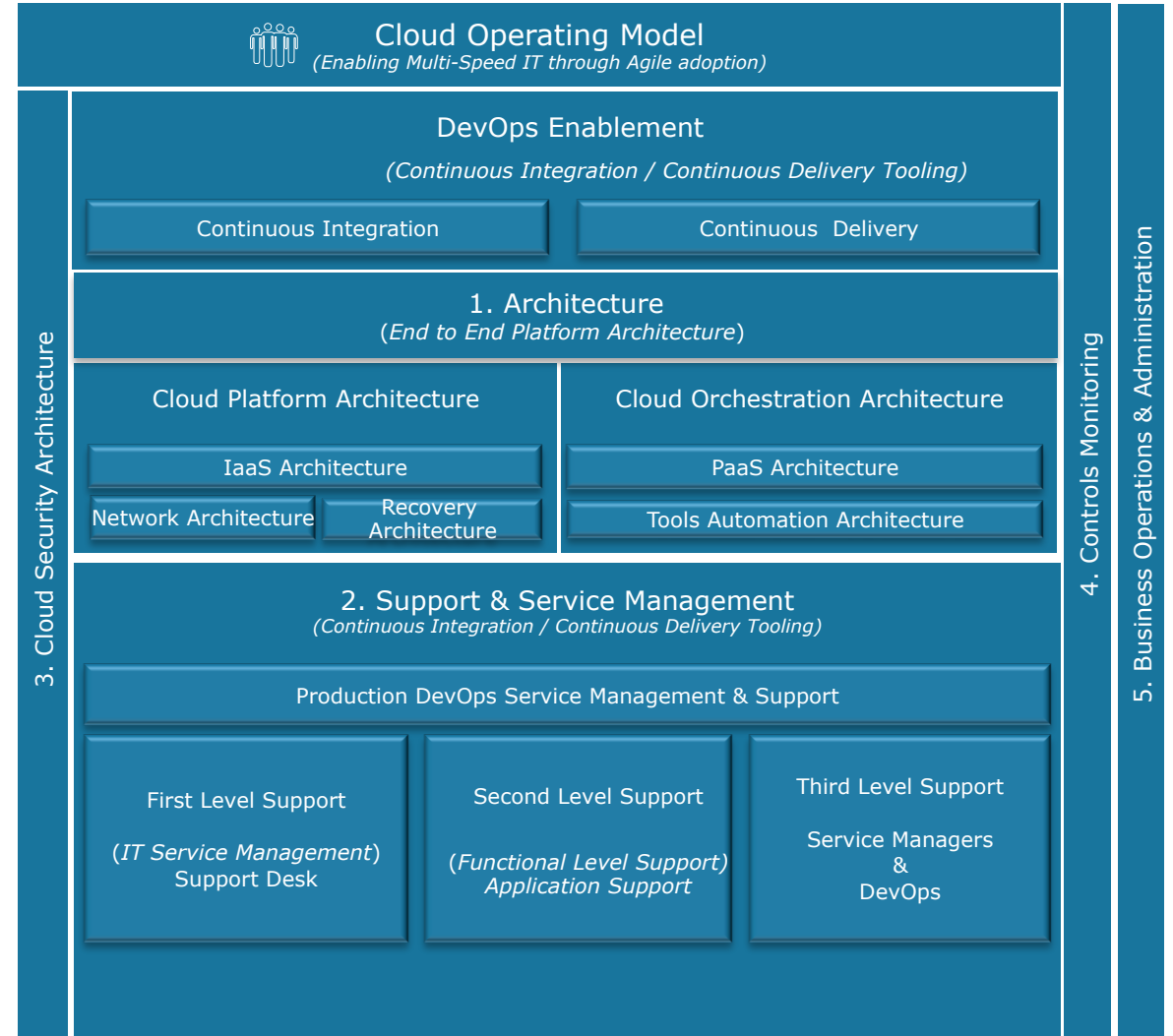
4. Controls Monitoring
An established security and controls monitoring function that ensures all controls are maintained and can stand up to control challenges
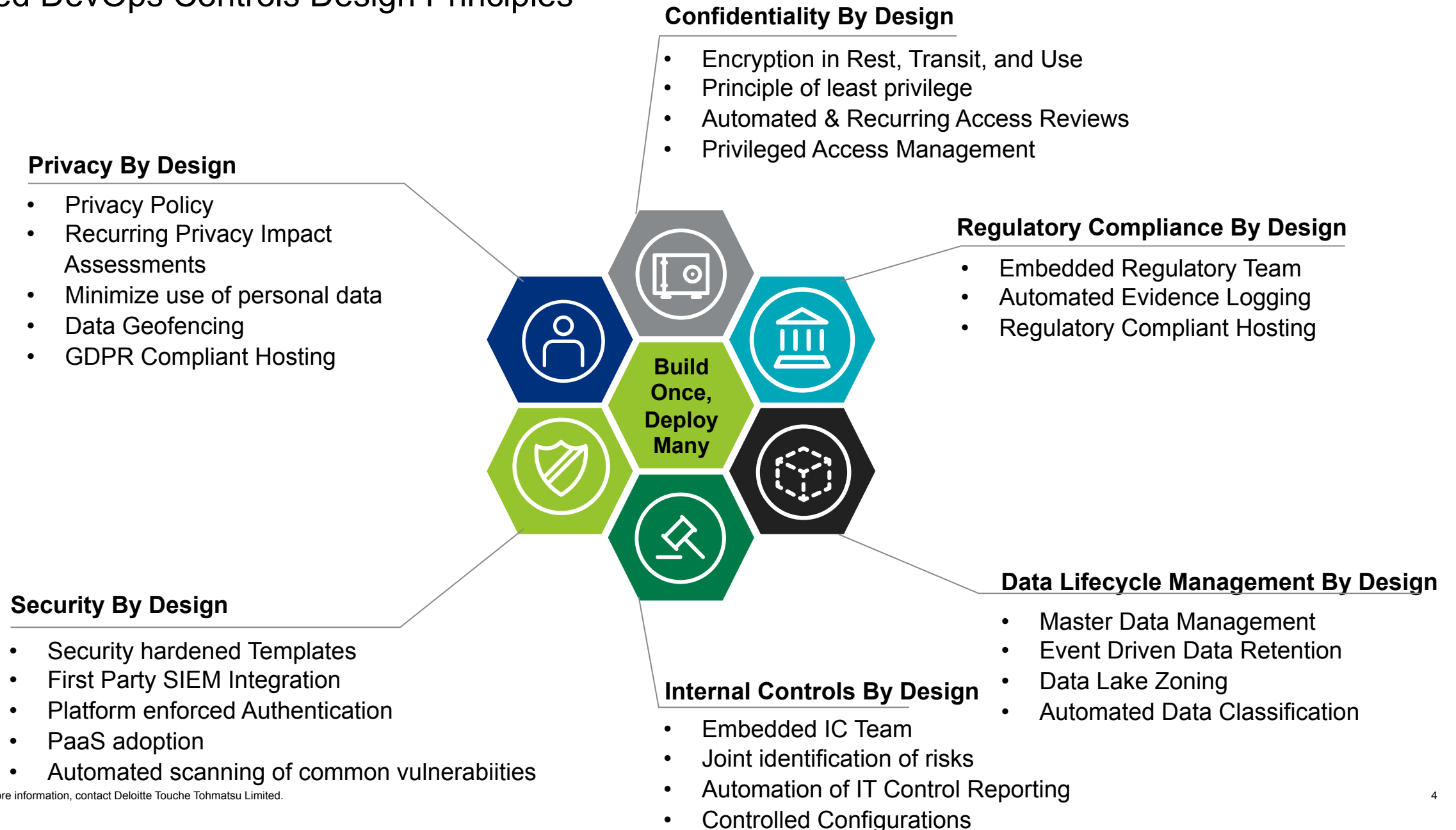
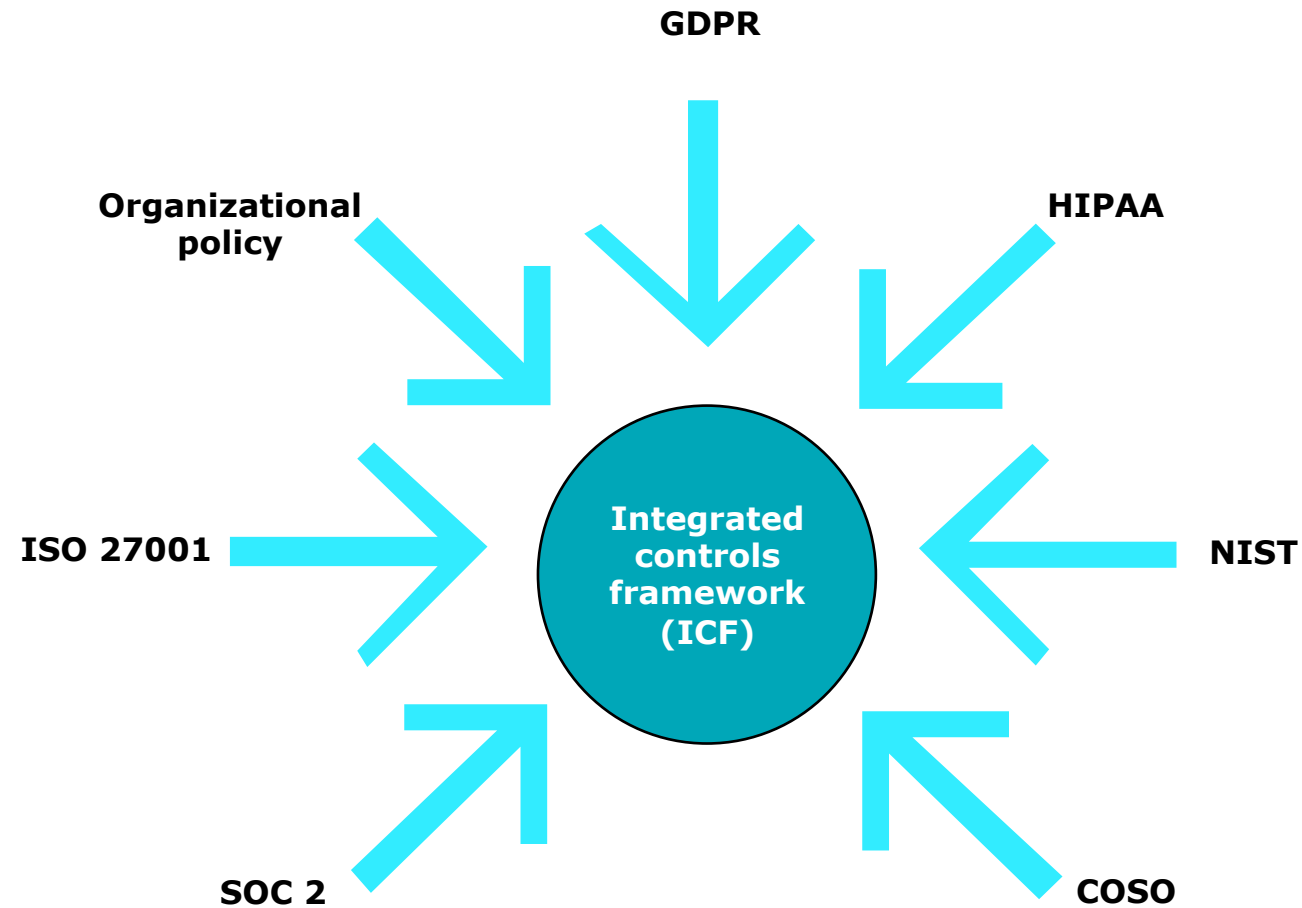**5. Business Operations & Administration**
An extensible accounting and administration function that ensure all usage and cloud cost are managed and maintained. Metering, billing and account management

---

Cloud Operating Model
(Enabling Multi-Speed IT through Agile adoption)

DevOps Enablement
(Continuous Integration / Continuous Delivery Tooling)

- Continuous Integration
- Continuous Delivery

1. Architecture
(End to End Platform Architecture)

Cloud Platform Architecture
- IaaS Architecture
- Network Architecture
- Recovery Architecture

Cloud Orchestration Architecture
- PaaS Architecture
- Tools Automation Architecture

2. Support & Service Management
(Continuous Integration / Continuous Delivery Tooling)

Production DevOps Service Management & Support

First Level Support
(IT Service Management) Support Desk

Second Level Support
(Functional Level Support) Application Support

Third Level Support
Service Managers & DevOps

3. Cloud Security Architecture

4. Controls Monitoring

5. Business Operations & Administration

Unified DevOps Controls Design Principles

Confidentiality By Design

- Encryption in Rest, Transit, and Use
- Principle of least privilege
- Automated & Recurring Access Reviews
- Privileged Access Management

Privacy By Design

- Privacy Policy
- Recurring Privacy Impact Assessments
- Minimize use of personal data
- Data Geofencing
- GDPR Compliant Hosting

Regulatory Compliance By Design

- Embedded Regulatory Team
- Automated Evidence Logging
- Regulatory Compliant Hosting

Build Once, Deploy Many

Security By Design

- Security hardened Templates
- First Party SIEM Integration
- Platform enforced Authentication
- PaaS adoption
- Automated scanning of common vulnerabiities

Internal Controls By Design

- Embedded IC Team
- Joint identification of risks
- Automation of IT Control Reporting
- Controlled Configurations

Data Lifecycle Management By Design

- Master Data Management
- Event Driven Data Retention
- Data Lake Zoning
- Automated Data Classification

Holistic software development lifecycle

Principles created via Integrated Controls Framework

- Access management
- Role-based provisioning
- Access reviews
- Data type sensitivities
- Wrap-up
- Operational monitoring

Stakeholder Management

Through partnership with DevOps, the IT Controls team can help navigate a complex web of stakeholder relationships.

With increased regulatory environments and focus on security, controls matters more to the business than ever before.

DevOps Compliance and Controls Myths

1) **It is not possible to be agile and comply with general IT controls**

   Investing in automation and 'controls by design' approaches make agile possible. By making these up front investments in culture and process, organizations find it easier to be agile and compliant.   By contrast, organizations that rely heavily upon manual approaches or leave controls/compliance to the end struggle.

2) **DevOps means 'Developers manage production'**

   This depends upon the needs of the organization.   For example, a financial regulator may need to see segregation of duties evidence, which influences organizational design. Other organizations may not face these requirements. What's most important is that responses to operational issues are cross functional, leveraging the best of both development and operations.

3) **A compliant DevOps organization can only be achieved in the cloud**

   While cloud technologies are an accelerator and enabler to a controlled/compliant organization, it's not the only way to achieve a controlled and compliant result. Organizations can adopt portions of the cloud to help with compliance, such as SaaS CI/CD tools and monitoring agents. Organizations can have their on-premises operations certified. Lastly, automation can be performed anywhere.

4) **Controls Slow down Continuous Integration/Continuous Deployment**

   Through environment segregation and release pipeline approvers, DevOps teams can work together so that development integrates and deploys code to pre-production environments at their own pace. It is important that controls are gradually introduced or enhanced

5) **A DevOps Model introduces organizational risk**

   The opposite is true. Repeatable, reusable processes are achieved by automation, replacing manual, error prone operations, reducing risk to organizations.

Questions?