Cloud Computing: What is Old is New Again

M. R. Pamidi, Ph. D.
Editor-in-Chief
IT Newswire

Executive Summary

Cloud Computing is one of the fastest-growing segments in IT, forecasted to reach US$235 billion globally by 2017.[1] There are hundreds of companies worldwide seeking to control supremacy by offering many "…as-a-service" products and services, often with very few differentiators except pricing which is in a downward spiral. C[3]DNA, a Silicon Valley startup, founded by professionals with deep industry experience and extensive academic background, offers an infrastructure-agnostic cloud product that could render many of today's 'leading' cloud vendors irrelevant in the not-too-distant future.

1. Introduction

Organizations moving to public clouds have a variety of well-established vendors to choose from. However, a major challenge they face is moving legacy enterprise applications to the cloud and still deliver predictable performance and security without re-architecting. This is because current public cloud Infrastructure-as-a-Service (IaaS) offerings are best suited to address the needs of new cloud-native applications architectures that have been designed to be stateless or use cloud native services and hence can scale out horizontally with the help of the right infrastructure. Unfortunately, this architecture heavily depends on infrastructure orchestration for managing application quality of service (QoS) and does not address enterprises' need to migrate and run legacy applications in the cloud without increasing complexity or vendor lock-in.

As industry veteran Kumar Malavalli, founder of Brocade, InMage, and other Silicon Valley startups, implied in 2009 (Figure 1), what needs to evolve is not a *cloud-native*, but a *cloud-agnostic*, globally interoperable, and massively scalable service creation, delivery, and assurance platform. This is because most mission-critical applications are multi-tiered and need to be operated and supported with high availability, low latency, optimal performance, regulatory compliance, and security—well into the foreseeable future.

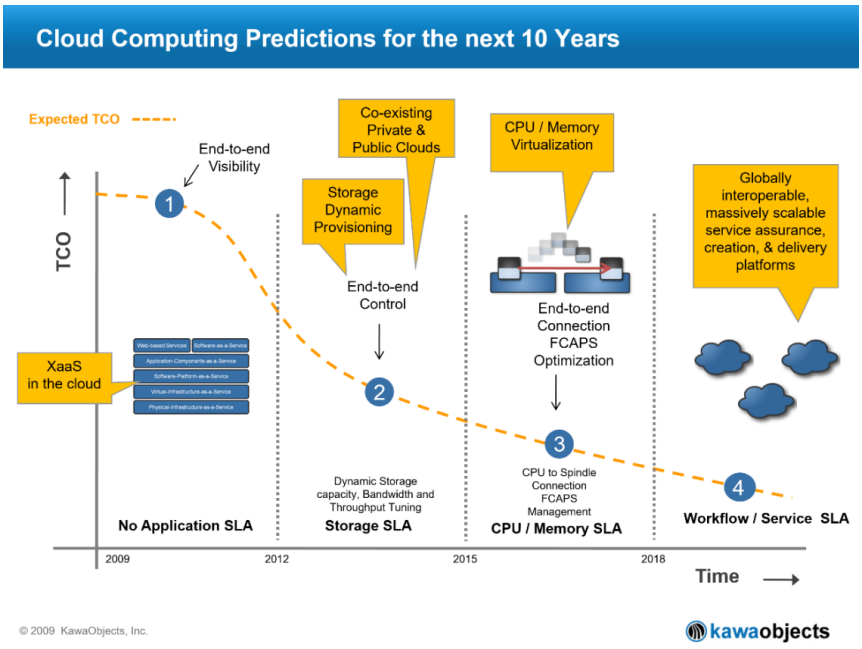

Figure 1. Vision for Next-Generation Cloud Computing[2]

[1] "IHS: Cloud Spending Will Top $235B by 2017," Dan Kobialka, April 7, 2014.
[2] "An Almost Silent Paradigm Shift Occurring in Information Technologies," Kumar Malavalli, Jan 29, 2016.

2. Problem with Today's Solutions

There is a wide variety of cloud vendors offering IaaS, PaaS, and SaaS in the market today. However, they all have certain limitations.

- Many are focused on just infrastructure provisioning: Azure, AWS, and Google can spin high-performance, low-latency distributed clusters in seconds; IBM SoftLayer can provision bare-metal clusters in 30 minutes; and Rackspace and Cisco Intercloud Services provide OpenStack VMs in minutes, if not seconds. However, they have to use a plethora of tools to gain visibility into application run-time changes in its resource requirements (CPU, memory, network bandwidth, latency, storage IOPs, throughput, and capacity).
- Even after getting visibility in an indirect fashion, application availability, performance and security are managed by controlling the infrastructure. This becomes more complicated when the application spans across multiple distributed infrastructures often owned and operated by different providers. The lack of application-centric self-management and portability, results in ever-growing cost and complexity.
- The infrastructure-centric application quality of service (QoS) control results in either vendor lock-in or increased complexity.

3. C^{3}DNA: A Breath of Fresh Air in a Cloud of Confusion

Are you confused about what to make of IaaS, PaaS, SaaS, DRaaS (disaster recovery), DBaaS (database), NSaaS (network services), SSaaS (security)…?

Figure 2. Blurred Vision of Cloud Computing

C^{3}DNA, a Silicon Valley-based startup, takes an entirely fresh approach to the problems cited above. One of their patents asserts that "the increase in the complexity of management of applications and security for virtualized and cloud-based resources is a symptom, not of software design or operational execution, but rather a fundamental architectural issue related to Gödel's prohibition of self-reflection in Turing machines."

C^{3}DNA's tactic goes back to the pioneering works of Alan Turing and von Neumann, considered the fathers of modern-day computing. Turing's model of computing is straightforward: Execute an algorithm using simple read, compute, and write instructions. von Neumann, *inter alia*, introduced the stored program computing model, often called the **von Neumann model** or **Princeton architecture**.

C^{3}DNA's core solution is based on a non-von Neumann implementation of a managed Turing machine, which founder Dr. Rao Mikkilineni developed in 2009, that exploits parallelism and agent architectures in computing nodes. Called Distributed Intelligent Managed Element (DIME), with its foundation based on the ground-breaking works of Turing, DIME integrates computational workflows with a parallel implementation of management workflows. This provides dynamic service management with auto-configuration, auto-scaling, self-repair, self-protection and live migration (self-* properties) of Linux processes with or without hypervisors, containers or

virtual machines (VMs). It eliminates VM image motion for providing mobility of applications required in current cloud implementations. It believes the end-to-end transaction visibility and control it offers introduces a new level of telecom-grade trust. For over a decade systems and network equipment vendors have been attempting to implement predictive, self-correcting, self-repairing, self-healing equipment with various degrees of success; C^{3}DNA is now delivering it to applications independent of the infrastructure in which they are executed.

C^{3}DNA's etymology is fascinating: The three 'C's are for cognition, computing, and communications. The '3' is for the three steps in the process: application 'DNA' captured by C^{3}DNA (policies, components, configuration, and workflow; the distributed DNA application control platform that interprets application DNA; and the agent-based mechanism that encapsulates and delivers this. DNA is DIME Network Architecture and was introduced as a new approach to handle dynamic aspects of cloud computing, namely, self-* properties, end-to-end service transaction security, and so on.

Cognitive computing refers to systems that learn at scale, reason with purpose, and interact with humans naturally. Rather than being explicitly programmed, they learn and reason from their interactions with us and from their experiences with their environment. So, to some extent, cognitive computing is *contextual computing*.

Even more interesting is the fact that Rao and his team's extensive telecom industry background has seeped into C^{3}DNA. As Kumar noted, using well-proven concepts from traditional telecommunications such as FCAPS (fault, configuration, accounting, performance, and security), out-of-band application control path, and a service routing and switching at application layer, it is possible now to provide self-awareness, self-configuration, self-monitoring, self-protection, and self-management properties to applications and workflows.

As Rao *et al*. clearly emphasize[3], all these self-* properties may be new to IT, but is old hat to the human body. Dyson notes that a recursive computing model in the genome enables the elegant unfolding of living organisms with self-configuration, self-monitoring, self-protection, and self-healing properties. "The analog of software in the living world is not a self-reproducing organism, but a self-replicating molecule of DNA. An elaborate self-extracting process restores entire directories of compressed genetic programs and reconstructs increasingly complicated levels of hardware on which the operating system runs."[4]

4. C^{3}DNA Product Highlights

In order to deliver mission-critical applications, enterprises currently rely on tightly integrated infrastructure-centric management tools and policies that result in vendor lock-in. Migrating applications to today's hyper-resilient commodity infrastructure requires a combination of application re-architecture, heavy use of VMs, and the implementation of myriad new operational management tools and orchestrators. C^{3}DNA's Cloud Equalizer is infrastructure-agnostic; with its self-reliant agent, users can dynamically discover, profile, and migrate distributed applications to any public, private, or hybrid cloud without changing the application, operating system, or re-architecting the application.

In order to achieve this, modern enterprise mission-critical cloud operations require:

- Policy-based migration of existing multi-tier or discrete applications along with constraints, dependencies and data without re-architecture – avoiding costly replication of VMs through VM image conversion or translation tools.
- Management of workload fluctuations to deliver predictable service levels for services running on a distributed infrastructure in datacenters and a hybrid, private, or public cloud environment.

[3] "The Turing O-Machine and the DIME Network Architecture: Injecting the Architectural Resiliency into Distributed Computing," Rao Mikkilineni *et al*., *Turing-100: The Alan Turing Centenary*, Andrei Voronkov (ed.), EPiC Series, vol. 10, June 22, 2012, pp. 239–251.

[4] *Darwin among the Machines, the evolution of global intelligence*, G. B. Dyson, Addison Wesley, Reading, MA, 1997.

- A common operational management model that is able to dynamically distribute workloads in order to achieve accounting benefit in terms of cost and budgeting while also ensuring data sovereignty, service performance, availability, regulatory compliance, and security.

C^{3}DNA's software is designed to describe algorithms or business processes in a precise form to be executed using CPU and memory. As previously discussed, Turing showed that any process that includes read, compute (change state), and write can be executed by a machine with finite states. However, this requires CPU and memory resources. The software does not know when resources are exhausted or what it needs. The hardware does not know what the software needs in terms of resources and to complete its intent. What C^{3}DNA does is to create a software meta-model in terms of its non-functional requirements (resource availability, performance, security, regulatory compliance, etc.) and a resource meta-model derived from the application meta-model. The resources are configured and the application is provisioned and monitored to ensure its intent is fulfilled. If deviations occur, the meta-knowledge of corrective actions in the form of policies is executed to correct the behavior.

This elegant approach allows C^{3}DNA to create application area networks (AANs) using management of each process and a signaling scheme that creates managers that manage groups with the group meta-model. This manager-of-manager process is recursive, just as companies organize themselves with a CEO, VP, Director, Supervisor, worker bee, etc. The softswitch allows each read/write to be influenced by external agents. This is what Turing introduced in his thesis as a Turing oracle modeled after the oracle of Delphi who knows the answers from outside the domain. Combining these concepts, C^{3}DNA has created AANs that configure, monitor, and control downstream subnetworks executing sub-processes. The meta-knowledge along with current monitored state allows preemptive action to control the evolution.

C^{3}DNA's HyperCloud Application Platform (Figures 3 and 4) enables:

- Enterprises to deploy multi-tier applications in their own datacenters and private, public, and hybrid clouds with complete Lifecycle Management.
- Migration and management of existing applications and new cloud-native applications to private, public, and hybrid clouds without depending on cloud-specific architectures or APIs.
- Predictable QoS across Federated Clouds without customer lock-in. C^{3}DNA uniquely addresses the inherent unreliability and unpredictability of "Distributed Infrastructure" and offers resource federation across distributed clouds without custom architectures and proprietary APIs.

5. C^{3}DNA Product Features and Benefits

Features	Benefits
• Infrastructure-agnostic application lifecycle QoS assurance • Policy-based or on-demand service, provisioning, self-repair, auto-scaling and true, live migration with transaction consistency without VM image motion • Non-functional requirements fulfillment with desired RPO and RTO • No dependence on infrastructure API for monitoring and orchestration of applications at run time	• No re-architecting for cloud and no changes to application or OS • Reduces VM sprawl significantly, resulting in an order of magnitude saving per VM • Inter-cloud and intra-cloud high availability, disaster recovery and optimized performance. • Easy and cost-effective integrations and migrations • Auto-scaling and self-healing result in significant OpEx reduction

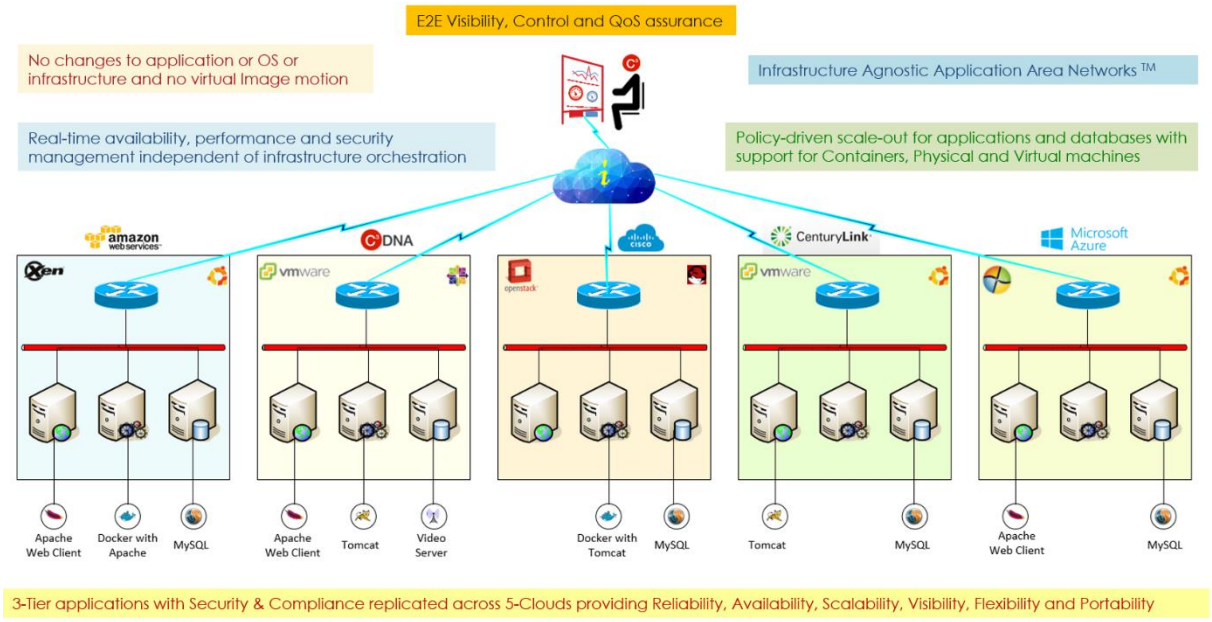

Figure 3. C[3]DNA service orchestration across private datacenter and multiple public clouds
Source: C[3]DNA

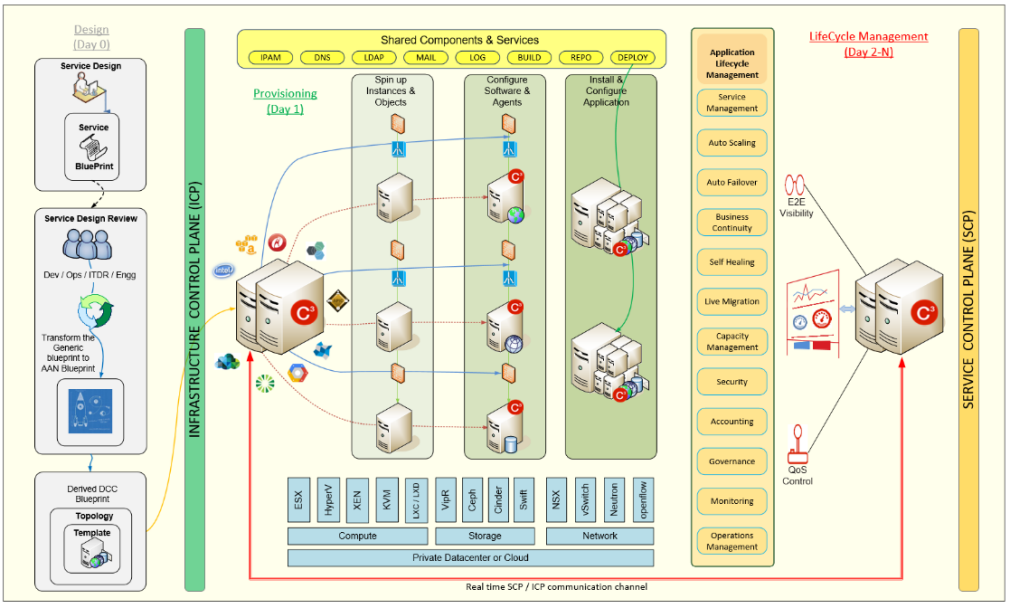

Figure 4. Infrastructure Provisioning and Service Orchestration and Provisioning
Source: C[3]DNA

6. Is C[3]DNA a Cloud Management Tool?

Cloud management comprises:

- Infrastructure provisioning (server nodes, network connections, and storage) using IaaS;
- Runtime provisioning of OS, application middleware, databases etc., using PaaS;
- Infrastructure monitoring to guess application availability, performance, security, and compliance fluctuations and needs; and
- Infrastructure orchestration to adjust the application QoS.

For example, some PaaS platforms, such as Apprenda, in the market consume and aggregate computational capacity and database servers from the foundational service catalog, aggregate them, and provide a policy-driven self-service consumption model for application deployment. They **do not** provision IaaS and assume that the resources are available.

CloudFoundry, for example, uses both IaaS and PaaS provisioning services to deploy the application and orchestrate the infrastructure for auto-scaling, failover, and migration for state preserving requirements. For stateless applications, it recreates them.

C^{3}DNA **does not** offer a cloud management system – hybrid, private, or public. Its solution is infrastructure-agnostic and it doesn't care how a cloud is managed as long as it provides us with a server with an operating system connected to a network and storage with the required SLAs and policies dictated by a Resource Blueprint which, in turn, is derived from an Application Blueprint. The Application Blueprint defines the non-functional requirements fulfilling topology and components along with dynamic policies that dictate application configuration, fault management, accounting management, performance management, and security management of the application. The only things the application cares about are node CPU, memory, network bandwidth and latency, storage IOPS, throughput, and capacity from the infrastructure. These are requested through and provided by the local OS in the node that hosts the application component. Comparing what C^{3}DNA does with cloud management systems is akin to comparing apples and oranges.

The following observations differentiate their approach from the current state of the art:

- Current cloud management solutions focus on application availability, performance, security, and compliance management by monitoring and controlling (or orchestrating) the infrastructure (servers, network, and storage). This increases the complexity when one starts to deploy applications in distributed clouds or infrastructure that is provided by different suppliers with different local infrastructure management practices. C^{3}DNA provides infrastructure agnostic application provisioning and orchestration using a dynamic control path overlay to infrastructure provisioning.
- A server/VM/container comprises infrastructure and is provisioned and managed differently in different clouds/infrastructures. Applications are always compiled to run in an OS and, therefore, C^{3}DNA's application abstraction above the OS is applicable in any infrastructure.
- Above all, none of the current technologies provide application-level read/write switching and routing which change the capabilities to allow the application itself as a network that is managed. This is what makes C^{3}DNA managed applications independent of infrastructure. You can have static infrastructure and mobile applications.
- C^{3}DNA's File Descriptor switching provides application HA/DR, application security, and application performance — independent of distributed infrastructure providers.
- They do not provide an application containerizer. They configure, monitor, and control applications using the OS services. They are more like a shell (UNIX bash or kron shell) where they sit between the external world and the OS.

Here is an analogy that works.

As soon as a telephone goes on hook (when a telephone number is dialed), the network knows the profile and makes decisions on whether this person is allowed to talk, has paid the bill, what kind of resources are needed, etc. As soon as the destination is dialed, the network knows the profile of the destination and sets up the right connections, monitors it, and assures its QoS. The telephone network does all these by itself and doesn't need to be told.

A network management system knows the nodes and links and their roles and optimizes them by repairing, reconfiguring, recombining, and replicating nodes, links, and sub-networks to ensure the goal is fulfilled. The network does all these by itself and doesn't need to be told.

Similarly, the Internet knows exactly how to connect two devices (IP addresses) and ensure its QoS. The Internet does all these by itself and doesn't need to be told.

What C^{3}DNA is doing is extending this to any application. As soon as an application goes on hook (executed), C^{3}DNA discovers its profile and configures, monitors, and manages it. C^{3}DNA gets its dependencies from the profile (connect a URL or connect a database) and sets up those connections, monitors them, and ensures QoS. C^{3}DNA does all these by itself and doesn't need to be told. Infrastructure management systems do not have this end-to-end view.

C^{3}DNA offers AAN that goes beyond current trends in providing application-aware or application-centric infrastructure. Area network denotes a group of nodes connected as a network and provides a vehicle for executing a service. AAN is a group of application components whose topology and resource requirements (CPU, memory, latency, bandwidth, storage IOPs, capacity, and throughput) are defined by the non-functional requirements. A network of such components is designed to deliver the functional requirements. Current state of the art locks these application components in various infrastructure nodes and orchestrates the infrastructure nodes to deliver the non-functional requirement fulfillment.

C^{3}DNA flips the paradigm by infusing self-awareness, self-configuration, self-monitoring, and policy-based self-control into application components and their management to comply with non-functional requirements. This allows C^{3}DNA to demonstrate features that current state of the art cannot. You can watch an interesting video at: http://www.c3dna.com/cognet.html

7. How is C^{3}DNA's Market Traction?

The company has raised $2 million in seed funding and has proof-of-concept or production environments deployed at CenturyLink, Cisco, Innova Solutions, Sabre Technologies, and Wells Fargo. It recently signed an agreement with the Indian IT company Tech Mahindra (NSE: TECHM) for the distribution of its Cloud Equalizer platform in Japan.

How do industry analysts and customers feel?

- "Blows my mind and challenges my understanding of IT." – *Hugh Ujhazy, Director, IDC*.
- "C^{3}DNA's Distributed Cognitive Transaction Platform will help us eliminate the vast majority of tools and their inherent cost and complexity." – *Gordon Tannura, SVP at Sabre Holdings*.
- "C^{3}DNA believes its approach – which takes a Turing machine 'oracle' as the design point – eliminates the need for infrastructure orchestration, and that firms that move VMs will be redundant in its model." – *451 Research*.

You can learn more about the company at its website.

8. Conclusions

To say we are very impressed with C^{3}DNA's product and technology is an understatement. The company's innovative and imaginative team has come up with a unique solution that could change the entire cloud computing vendor landscape. As one industry analyst put it, C^{3}DNA is Uber-ifying this industry that could spell the end for many vendors still fighting the IaaS, PaaS, SaaS, XaaS…wars.

