Open access Journal International Journal of Emerging Trends in Science and Technology

COMMUNITY CLOUD COMPUTING USING RESTFUL API

Authors
Dhivya.M, Divya.K, Ramesh Shahabadkar
Department of Computer Science
Christ Uniersity, Bangalore
Email: *dhivya.m@mtech.christuniversity.in, divya.k@mtech.christuniversity.in*
ramesh.shahabadkar@christuniversity.in

ABSTRACT:
Cloud Computing is rising fast, with its data centres growing at an unprecedented rate. However, this has come with concerns over privacy, efficiency at the expense of resilience, and environmental sustainability, because of the dependence on Cloud vendors such as Google, Amazon and Microsoft. Our response is an alternative model for the Cloud conceptualisation, providing a paradigm for Clouds in the community, utilising networked personal computers for liberation from the centralised vendor model. Community Cloud Computing (C3) offers an alternative architecture, created by combing the Cloud with paradigms from Grid Computing, principles from Digital Ecosystems, and sustainability from Green Computing, while remaining true to the original vision of the Internet. Here, we have introduced a suitable mechanism for community currency and restful clouds using API in c3 to increase the availability in vendors cloud. It is more technically challenging than Cloud Computing, having to deal with distributed computing issues, including heterogeneous nodes, varying quality of service, and additional security constraints.

Index Terms: *Cloud Computing, Community Cloud, Community Cloud Computing, Green Computing, Sustainability and Restful cloud using API*

INTRODUCTION

The recent development of Cloud Computing provides a compelling value proposition for organisations to outsource their Information and Communications Technology (ICT) infrastructure [1]. However, there are growing concerns over the control ceded to large Cloud vendors [2], especially the lack of information privacy [3]. Computing are growing exponentially [4], creating an ever-increasing *carbon footprint* and therefore raising environmental concerns [5], [6]. Also, the data centres required for Cloud.

The distributed resource provision from Grid Computing, distributed control from Digital Ecosystems, and sustainability from Green Computing, can remedy these concerns. So, Cloud Computing combined with these approaches would provide a compelling socio-technical conceptualisation for sustainable distributed computing, utilising the spare resources of networked personal computers collectively to provide the facilities of a virtual *data centre* and form a Community Cloud. Many consider APIs to be the best method for organizations to access services provided by cloud computing vendors. Cloud consumers use APIs as software interfaces to connect and consume resources in various ways, though the optimal or contemporary route is to use a Restful protocol-based API.

Read on to gain an understanding of what APIs are and how they are used —particularly when it comes to REST APIs and cloud computing services Therefore, essentially reformulating the Internet to reflect its current uses and scale, while maintaining the original intentions [7] for sustainability in the face of adversity. Including extra capabilities embedded into the infrastructure which would become as fundamental and invisible as moving packets is today.

CLOUD COMPUTING

Cloud Computing is the use of Internet-based technologies for the provision of services [1], originating from the *cloud* as a metaphor for the Internet, based on depictions in computer network diagrams to abstract the complex infrastructure it conceals [8]. It can also be seen as a commercial evolution of the academic-oriented Grid Computing [9] , succeeding where Utility Computing struggled [10], [11] , while making greater use of the self-management advances of Autonomic Computing [12]. It offers the illusion of infinite computing resources available on demand, with the elimination of upfront commitment from users, and payment for the use of computing resources on a short-term basis as needed [3]. Furthermore, it does not require the node providing a service to be present once its service is deployed [3]. It is being promoted as the cutting-edge of scalable web application development [3], in which dynamically scalable and often virtualised resources are provided as a service over the Internet [13], [1], [14] , [15], with users having no knowledge of, expertise in, or control over the technology infrastructure of the Cloud supporting them [16]. It currently has significant momentum in two extremes of the web development industry [3] , [1]: the consumer web technology incumbents who have resource surpluses in their vast *data centres*[1], and various consumers and start-ups that do not have access to such computational resources. Cloud Computing conceptually incorporates Software-as-a-Service (SaaS) [18], Web 2.0 [19] and other technologies with reliance on the Internet, providing common business applications online through web browsers to satisfy the computing needs of users, while the software and data are stored on the servers. Figure 1 shows the typical configuration of Cloud Computing at run-time when consumers visit an application served by the central Cloud, which is housed in one or more data centres [20]. *Green* symbolises resource *consumption*, and *yellow* resource *provision*. The role of *coordinator* for resource provision is designated by *red*, and is centrally controlled. Even if the central node is implemented as a distributed grid, which is the usual incarnation of a data centre, control is still centralised. Providers, who are the controllers, are usually companies with other web activities that require large computing

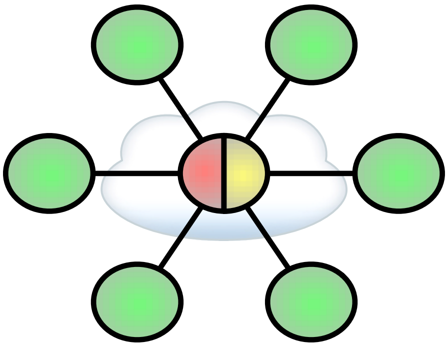


Figure1. Cloud Computing: Typical configuration when consumers visit an application served by the central Cloud, which is housed in one or more data centres [20]

Green symbolises resource consumption, and yellow resource provision. The role of coordinator for resource provision is designated by red, and is centrally controlled. Resources, and in their efforts to scale their primary businesses have gained considerable expertise and hardware. For them, Cloud Computing is a way to resell these as a new product while expanding into a new market. Consumers include everyday users, Small and Medium sized Enterprises (SMEs), and ambitious start-ups whose innovation potentially threatens the incumbent providers.

---

[1] A *data centre* is a facility, with the necessary security devices and environmental systems (e.g. air conditioning and fire suppression), for housing a *server farm*, a collection of computer servers that can accomplish server needs far beyond the capability of one machine [17].

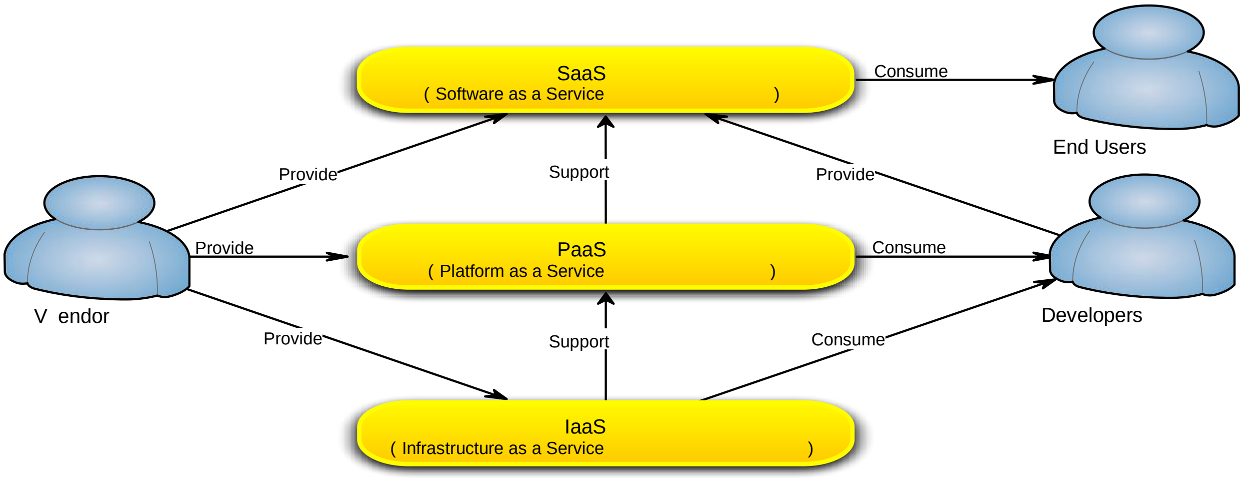


Figure 2. Abstractions of Cloud Computing: While there is a significant

A. Layers of Abstraction

While there is a significant *buzz* around Cloud Computing, there is little clarity over which offerings qualify or their interrelation. The key to resolving this confusion is the realisation that the various offerings fall into different levels of abstraction, as shown in Figure 2, aimed at different market segments.

Infrastructure-as-a-Service (IaaS) [21]: At the most basic level of Cloud Computing offerings, there are providers such as Amazon [22] and Mosso [23], who provide *machine instances* to developers. These instances essentially behave like dedicated servers that are controlled by the developers, who therefore have full responsibility for their operation. So, once a machine reaches its performance limits, the developers have to manually instantiate another machine and scale their application out to it. This service is intended for developers who can write arbitrary software on top of the infrastructure with only small compromises in their development methodology.

Platform-as-a-Service (PaaS) [24]: One level of abstraction above, services like Google App Engine [25] provide a programming environment that abstracts machine instances and other technical details from developers. The programs are executed over data centres, not concerning the developers with matters of allocation. In exchange for this, the developers have to handle some constraints that the environment imposes on their application design, for example the use of *key-value stores*[2]instead of *relational databases*.

3.Software-as-a-Service (SaaS) [18]: At the consumer-facing level are the most popular examples of Cloud Computing, with well-defined applications offering buzz around Cloud Computing, there is little clarity over which offerings qualify or their interrelation. The key to resolving this confusion is the realisation that the various offerings fall into different levels of abstraction, aimed at different market segments.users online resources and storage. This differentiates SaaS from traditional websites or web applications which do not interface with user information (e.g. documents) or do so in a limited manner. Popular examples include Microsoft's (Windows Live) Hotmail, office suites such as Google Docs and Zoho, and online business software such as Salesforce.com.To better understand Cloud Computing we can categorise the roles of the various actors. The *vendor* as resource provider has already been discussed. The application *developers* utilise the resources provided, building services for the *end users*. This separation of roles helps define the stakeholders and their differing interests. However, actors can take on multiple roles, with *vendors* also developing services for the *end users*, or *developers* utilising the services of others to build their own

---

[2] A distributed storage system for structured data that focuses on scalability, at the expense of the other benefits of relational databases [26], e.g. Google's BigTable [27] and Amazon's SimpleDB [28].

services. Yet, within each Cloud the role of provider, and therefore controller, can only be occupied by the *vendor* providing the Cloud.

B. Concerns

The Cloud Computing model is not without concerns, as others have noted [29], [3], and we consider the following as primary:

Failure of Monocultures: The uptime[3] of Cloud Computing based solutions is an advantage, when compared to businesses running their own infrastructure, but often overlooked is the co-occurrence of downtime in vendor-driven *monocultures*. The use of globally decentralised *data centres* for vendor Clouds minimises failure, aiding its adoption. However, when a cloud fails, there is a cascade effect crippling all organisations dependent on that Cloud, and all those dependent upon them. This was illustrated by the Amazon (S3) Cloud outage [31] , which disabled several other dependent businesses. So, failures are now system-wide, instead of being partial or localised. Therefore, the efficiencies gained from centralising infrastructure for Cloud Computing are increasingly at the expense of the Internet's resilience.

Convenience vs Control: The growing popularity of Cloud Computing comes from its convenience, but also brings vendor control, an issue of ever-increasing concern. For example, Google Apps for in-house e-mail typically provides higher uptime [32], but its failure [33] highlights the issue of lock-in that comes from depending on vendor Clouds. The even greater concern is the loss of information privacy, with vendors having full access to the resources stored on their Clouds. So much so the British government is considering a 'G Cloud' for government business applications [34]. In particularly sensitive cases of SMEs and start-ups, the provider-consumer relationship that Cloud Computing fosters between the owners of resources and their users could potentially be detrimental, as there is a potential conflict of interest for the providers. They profit by providing resources to up-and-coming players, but also wish to maintain dominant positions in their consumer facing industries.

Environmental Impact: The other major concern is the ever-increasing *carbon footprint* from the *exponential growth* [4] of the *data centres* required for Cloud Computing. With the industry expected to exceed the airline industry by 2020 [6], raising sustainability concerns [5]. The industry is being motivated to address the problem by legislation [6], [35], the operational limit of power grids (being unable to power anymore servers in their data centres) [36], and the potential financial benefits of increased efficiency [37], [6]. Their primary solution is the use of *virtualisation*[4] to maximise resource utilisation [39], but the problem remains [40], [41].

While these issues are endemic to Cloud Computing, they are not flaws in the Cloud conceptualisation, but the vendor provision and implementation of Clouds [25], [22] , [42]. There are attempts to address some of these concerns, such as a portability layer between vendor Clouds to avoid lock-in [43]. However, this will not alleviate issues such as inter-Cloud latency [44]. An open source implementation of the Amazon (EC2) Cloud [22], called Eucalyptus [45], allows a data centre to execute code compatible with Amazon's Cloud. Allowing for the creation of *private internal* Clouds, avoiding vendor lock-in and providing information privacy, but only for those with their own data centre and so is not really Cloud Computing (which by definition is to avoid owning data centres [1]). Therefore, vendor Clouds remain synonymous with Cloud Computing [13], [1], [14], [15]. Our response is an alternative model for the Cloud conceptualisation, created by combining the Cloud with paradigms from Grid Computing, principles from Digital Ecosystems, and sustainability from Green Computing, while remaining true to the original vision of the Internet [46].

---

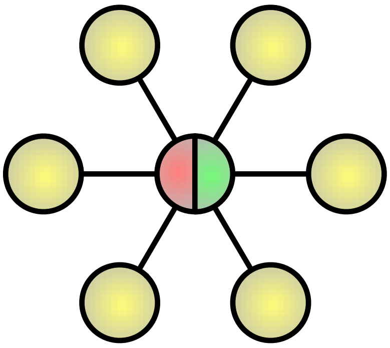
[4] Virtualisation is the creation of a virtual version of a resource, such as a server, which can then be stored, migrated, duplicated, and instantiated as needed, improving scalability and work load management [38].

GRID COMPUTING: DISTRIBUTING PROVISION

Grid Computing is a form of distributed computing in which a *virtual super computer* is composed from a cluster Figure 3. Grid Computing: Typical configuration in which resource provision is managed by a group of distributed nodes [47]. Green symbolises resource consumption, and yellow resource provision. The role of coordinator for resource provision is designated by red, and is centrally controlled of networked, loosely coupled computers, acting in concert to perform very large tasks [47]. It has been applied to computationally intensive scientific, mathematical, and academic problems through volunteer computing, and used in commercial enterprise for such diverse applications as drug discovery, economic forecasting, seismic analysis, and back-office processing to support e-commerce and web services [47].What distinguishes Grid Computing from *cluster computing* is being more loosely coupled, heterogeneous, and geographically dispersed [47]. Also, grids are often constructed with general-purpose grid software libraries and middleware, dividing and apportioning pieces of a program to potentially thousands of computers [47]. However, what distinguishes Cloud Computing from Grid Computing is being web-centric, despite some of its definitions being conceptually similar (such as computing resources being consumed as electricity is from power grids) [9].

DIGITAL ECOSYSTEMS: DISTRIBUTING CONTROL

Digital Ecosystems are distributed adaptive open sociotechnical systems, with properties of self-organisation, scalability and sustainability, inspired by natural ecosystems [48], [49]. Emerging as a novel approach to the catalysis of sustainable regional development driven by SMEs [50]. Aiming to help local economic actors become active players in globalisation [51], valorising their local culture and vocations, and enabling them to interact and create value networks at the global level [52].



Increasingly this approach, dubbed *globalisation*, is being considered a successful strategy of globalisation that preserves regional growth and identity [53], [54], [55], and has been embraced by the mayors and decision-makers of thousands of municipalities [56]. The community focused on the deployment of Digital Ecosystems, Regions for Digital Ecosystems Network (REDEN) [50], is supported by projects such as the Digital Ecosystems Network of regions for (4) Dissemination and Knowledge Deployment (DEN4DEK) [57]. This thematic network that aims to share experiences and disseminate knowledge to let regions effectively deploy of Digital Ecosystems at all levels (economic, social, technical and political) to produce real impacts in the economic activities of European regions through the improvement of SME business environments.

In a traditional market-based economy, made up of sellers and buyers, the parties exchange property, while in a new network-based economy, made up of servers and clients, the parties share access to services and experiences [58]. Digital Ecosystems aim to support network-based economies reliant on next-generation ICT that will extend the Service-Oriented Architecture (SOA) concept [59] with the automatic combining of available and applicable services in a scalable architecture, to meet business user requests for applications that facilitate business processes. Digital Ecosystems research is yet to consider scalable resource provision, and therefore risks being subsumed into vendor Clouds at the infrastructure level, while striving for decentralisation at the service level. So, the realisation of their vision requires a form of Cloud Computing, but with their principle of community-based infrastructure where individual users share ownership [48].

GREEN COMPUTING: GROWING SUSTAINABLY

Green Computing is the efficient use of computing resources, with the primary objective being to account for the *triple bottom line*[5], an expanded spectrum of values and criteria for measuring organisational (and societal) success [61]. Given computing systems existed before concern over their environmental impact, it has generally been implemented retroactively, but is now being considered at the development phase [61]. It is systemic in nature, because ever-increasingly sophisticated modern computer systems rely upon people, networks and hardware. So, the elements of a *green* solution may comprise items such as end user satisfaction, management restructuring, regulatory compliance, disposal of electronic waste, telecommuting, virtualisation of server resources, energy use, thin client solutions and return on investment [61].

One of the greatest environmental concerns of the industry is their data centres [41], which have increased in number over time as business demands have increased, with facilities housing a rising amount of evermore powerful equipment [17]. As data centres run into limits related to power, cooling and space, their ever-increasing operation has created a noticeable impact on power grids [36]. To the extent that data centre efficiency has become an important global issue, leading to the creation of the Green Grid [62], an international non-profit organisation mandating an increase in the energy efficiency of data centres. Their approach, virtualisation, has improved efficiency [40], [41], but is optimising a flawed model that does not consider the whole system, where resource provision is disconnected from resource consumption. For example, competing vendors must host significant redundancy in their data centres to manage usage spikes and maintain the illusion of infinite resources. So, we would argue that an alternative more systemic approach is required, where resource consumption and provision are connected, to minimise the environmental impact and allow sustainable growth.

COMMUNITY CLOUD

C3 arises from concerns over Cloud Computing, specifically control by vendors and lack of environmental sustainability. The Community Cloud aspires to combine distributed resource provision from Grid Computing, distributed control from Digital Ecosystems and sustainability from Green Computing, with the use cases of Cloud Computing, while making greater use of self-management advances from Autonomic Computing. Replacing vendor Clouds by shaping the underutilised resources of user machines to form a Community Cloud, with nodes potentially fulfilling all roles, *consumer*, *producer*, and most importantly *coordinator*, as shown in Figure 4.

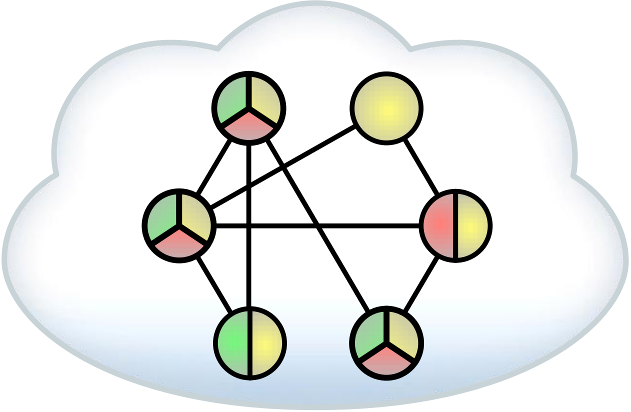


Figure 4. Community Cloud: Created from shaping the underutilised resources of user machines, with nodes potentially fulfilling all roles, consumer, producer, and most importantly coordinator. Green symbolises resource consumption, yellow resource provision, and red resource coordination.

A. Conceptualisation

The conceptualisation of the Community Cloud draws upon Cloud Computing [20], Grid Computing [9], Digital Ecosystems [48], Green Computing [63] and Autonomic Computing [12]. A paradigm for Cloud Computing in the *community*, without dependence on Cloud vendors, such as Google, Amazon, or Microsoft.

---

[5] The triple bottom line (*people, planet, profit*) [60].

Openness: Removing dependence on vendors makes the Community Cloud the open equivalent to vendor Clouds, and therefore identifies a new dimension in the open versus proprietary struggle [64] that has emerged in code, standards and data, but has yet to be expressed in the realm of hosted services.

Community: The Community Cloud is as much a social structure as a technology paradigm [65], because of the community ownership of the infrastructure. Carrying with it a degree of economic scalability, without which there would be diminished competition and potential stifling of innovation as risked in vendor Clouds.

Individual Autonomy: In the Community Cloud, nodes have their own utility functions in contrast with data centres, in which dedicated machines execute software as instructed. So, with nodes expected to act in their own selfinterest, centralised control would be impractical, as with consumer electronics like game consoles [66]. Attempts to control user machines counter to their self-interest results in cracked systems, from black market hardware modifications and *arms races* over hacking and securing the software (routinely lost by the vendors) [66]. In the Community Cloud, where no concrete vendors exist, it is even more important to avoid antagonising the users, instead embracing their self interest and harnessing it for the benefit of the community with measures such as a *community currency*.

Identity: In the Community Cloud each user would inherently possess a unique identity, which combined with the structure of the Community Cloud should lead to an inversion of the currently predominant membership model. So, instead of users registering for each website (or service) anew, they could simply add the website to their identity and grant access. Allowing users to have multiple services connected to their identity, instead of creating new identities for each service. This relationship is reminiscent of recent application platforms, such as Facebook's f8 and Apple's App Store, but decentralised in nature and so free from vendor control. Also, allowing for the reuse of the connections between users, akin to Google's Friend Connect, instead of re-establishing them for each new application.

Graceful Failures: The Community Cloud is not owned or controlled by any one organisation, and therefore not dependent on the lifespan or failure of any one organisation. It therefore ought be robust and resilient to failure, and immune to the system-wide cascade failures of vendor Clouds, because of the diversity of its supporting nodes. When occasionally failing doing so gracefully, non-destructively, and with minimal downtime, as the unaffected nodes mobilise to compensate for the failure.

Convenience and Control: The Community Cloud, unlike vendor Clouds, has no inherent conflict between convenience and control, resulting from its community ownership providing distributed control, which would be more democratic. However, whether the Community Cloud can provide technically quality equivalent or superior to its centralised counterparts is an issue that will require further research.

Community Currency: The Community Cloud would require its own currency to support the sharing of resources, a *community currency*, which in economics is a medium (currency), not backed by a central authority (e.g. national government), for exchanging goods and services within a community [67]. It does not need to be restricted geographically, despite sometimes being called a local currency [68]. An example is the Fureai kippu system in Japan, which issues credits in exchange for assistance to senior citizens [69]. Family members living far from their parents can earn credits by offering assistance to the elderly in their local community, which can then be transferred to their parents and redeemed by them for local assistance [69].

Quality of Service: Ensuring acceptable quality of service (QoS) in a heterogeneous system will be a challenge. Not least because achieving and maintaining the different aspects of QoS will require reaching *critical mass* in the participating nodes and available services. Thankfully, the *community currency* could support long-term promises by resource providers and allow the higher quality providers, through market forces, to command a higher price for their service provision. Interestingly, the Community Cloud could provide a better QoS than vendor Clouds, utilising time-based and geographical variations advantageously in the dynamic scaling of resource provision.

Environmental Sustainability: We expect the Community Cloud to have a smaller *carbon footprint* than vendor Clouds, on the assumption that making use of underutilised user machines requires less energy than the dedicated data centres required for vendor Clouds. The server farms within data centres are an intensive form of computing resource provision, while the Community Cloud is more organic, growing and shrinking in a symbiotic relationship to support the demands of the community, which in turn supports it.

Service Composition: The great promise of serviceoriented computing is that the *marginal cost* of creating the n-th application will be virtually zero, as all the software required already exists to satisfy the requirements of other applications. Only their composition and orchestration are required to produce a new application [70], [71].

Within vendor Clouds it is possible to make services that expose themselves for composition and compose these services, allowing the hosting of a complete service-oriented architecture [20]. However, current service composition technologies have not gained widespread adoption [72]. Digital Ecosystems advocate service composability to avoid centralised control by large service providers, because easy service composition allows coalitions of SMEs to compete simply by composing simpler services into more complex services that only large enterprises would otherwise be able to deliver [52]. So, we should extend decentralisation beyond resource provision and up to the service layer, to enable service composition within the Community Cloud. *B. Architecture*

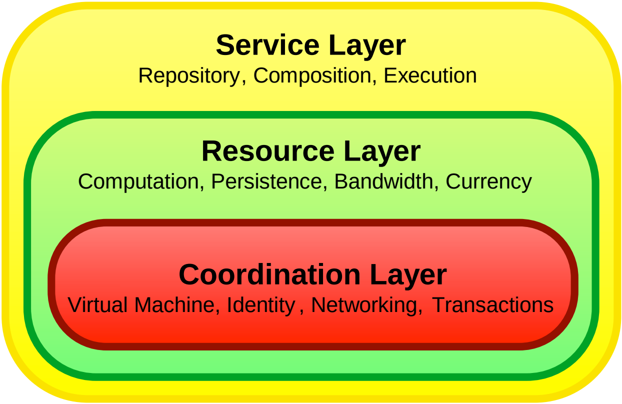


Figure 5. Community Cloud Computing: An architecture in which the most fundamental layer deals with distributing coordination. One layer above, resource provision and consumption are arranged on top of the coordination framework. Finally, the service layer is where resources are combined into end-user accessible services, to then themselves be composed into higher-level services.

The method of materialising the Community Cloud is the distribution of its server functionality amongst a population of nodes provided by user machines, shaping their underutilised resources into a *virtual data centre*. While straightforward in principle, it poses challenges on many different levels. So, an architecture for C3 can be divided into three layers, dealing with these challenges iteratively. The most fundamental layer deals with distributing *coordination*, which is taken for granted in homogeneous data centres where good connectivity, constant presence and centralised infrastructure can be assumed. One layer above, *resource* provision and consumption are arranged on top of the coordination framework. Easy in the homogeneous grid of a data centre where all nodes have the same interests, but more challenging in a distributed heterogeneous environment. Finally, the *service* layer is where resources are combined into end-user accessible services, to then themselves be composed into higher-level services. Complementary currencies that are available, and choose the type or types that suit your needs best.

IN THE COMMUNITY CLOUD

While we have covered the fundamental motivations and architecture of the Community Cloud, its practical application may still be unclear. So, this section discusses the cases of Wikipedia and YouTube, where the application of C3 would yield significant benefits, because they have unstable funding models, require increasing scalability, and are community oriented.

A.Wikipedia

Wikipedia suffers from an ever-increasing demand for resources and bandwidth, without a stable supporting revenue source [109]. Their current funding model requires continuous monetary donations for the maintenance and expansion of their infrastructure [110]. The alternative being contentious advertising revenues [109], which caused a long-standing conflict within their community [111]. While it would provide a more scalable funding model, some fear it would compromise the content and/or the public trust in the content [112]. Alternatively, the Community Cloud could provide a self-sustaining scalable resource provision model, without risk of compromising the content or public trust in the content, because it would be compatible with their communal nature (unlike their current *data centre* model), with their user base accomplishing the resource provision they require.

Were Wikipedia to adopt C3, it would be distributed throughout the Community Cloud alongside other services. With the core operations of Wikipedia, providing webpages and executing server-side scripts, being handled as service requests. Participants would use their *community currency* to interact with Wikipedia, performing a search or retrieving a page, while gaining *community currency* for helping to host Wikipedia across the Community Cloud. More complicated tasks, such as editing a Wikipedia webpage, would require an update to the distributed storage of the Community Cloud, achieved by transmitting the new data through its network of nodes, most likely using an eventual consistency model [90].

YouTube

YouTube requires a significant bandwidth for content distribution, significant computational resources for video transcoding, and is yet to settle on a profitable business model [113], [114]. In the Community Cloud, websites like YouTube would also have a self-sustaining scalable resource provision model, which would significantly reduce the income required for them to turn a profit.

Were YouTube to adopt C3, it would also be distributed throughout the Community Cloud alongside other services. Updates such as commenting on a YouTube video, would similarly need to propagate through the distributed persistence layer. So, the community would provide the bandwidth for content distribution, and the computational resources for video transcoding, required for YouTube's service. The QoS requirements for YouTube are significantly different to those of Wikipedia, because while constant throughput is desirable for video streaming, occasional packet loss is tolerable. Also, YouTube's streaming of live events has necessitated the services of bespoke content distribution networks [115], a type of service for which the Community Cloud would naturally excel.

We have discussed Wikipedia and YouTube in the Community Cloud, but other sites such as arXiv and Facebook would equally benefit. As C3's organisational model for resource provision moves the cost of service provision to the user base, effectively creating a micro-payment scheme, which would dramatically lower the barrier of entry for innovative start-ups.

C. Restful APIs

Restful APIs follow the SOA model and, therefore, are often used by web service-based software architectures via XML or JSON for integration purposes. This means that they are consumed via an Internet browser or by web servers. As mentioned before, Restful APIs are a relatively new technology requiring developers to have a thorough knowledge of current Web 2.0 technologies. Therefore, many organizations lacking an impetus to update their technology stack may not have the option of using this offering. Examples of organizations that may not be focused on Restful APIs these days would certainly overlap with those not looking to become cloud consumers. These may include law firms, higher education institutions, and nonprofit groups. However, Restful APIs are a reality that is here to stay.

Technology has come full circle in the sense that the majority of processing and storage resources have been pushed away from the end device, much like the days when mainframes and dumb terminals were prevalent. This is due to globalization, an increasingly remote or mobile workforce, ubiquitous Internet connectivity, and the maturity of SOA and Web 2.0 technologies such as RESTful APIs. With the addition of cloud computing used for data processing tasks via APIs, this trend is moving toward critical mass.

Cloud Computing and APIs

Cloud computing services by nature are distributed. So, the use of web-based RESTful APIs by consumers is a logical solution for the remote consumption of data processing services. And, when economic downturns, emerging markets, and lower barriers to entry for competitors are brought to mind, it is not hard to understand why there is a renewed focus on technology for competitive advantage.

REST APIs are essential as businesses try to gain or keep a foothold in their industries or markets while dealing with globalization, an increasingly remote or mobile workforce, and ubiquitous Internet connectivity. These realities have created vastly different end-user requirements that need to be satisfied. The cloud is helping businesses realize these new priorities. However, for an organization to properly use the cloud, the technology staff should understand the nuances of cloud computing, as well as some common use cases for using the cloud or cloud service provider (CSP) API solutions.

CONCLUSIONS

We have presented the Community Cloud as an alternative to Cloud Computing, created from blending its usage scenarios with paradigms from Grid Computing, principles from Digital Ecosystems, self-management from Autonomic Computing ,sustainability from Green Computing and restful API. So, C3 utilises the spare resources of networked personal computers to provide the facilities of data centres, such that the community provides the computing power for the Cloud they wish to use. A socio-technical conceptualisation for sustainable distributed computing. While the Open Cloud Manifesto is well intentioned, its promotion of open standards for vendor Cloud interoperability has proved difficult . RESTful Clouds is introduced to encourage the availability in vendors cloud. The addition of cloud computing used for data processing tasks via APIs, this trend is moving toward critical mass. The Community Cloud has encouraged innovation in vendor Clouds, forming a relationship analogous to the creative tension between open source and proprietary software. REST APIs are essential as businesses try to gain or keep a foothold in their industries or markets while dealing with globalization, an increasingly remote or mobile workforce, and ubiquitous Internet connectivity. We have refined the various elements of C3, such as suitable mechanisms for a *community currency*.

REFERENCES

1. M. Haynie, “Enterprise cloud services: Deriving business value from Cloud Computing,” Micro Focus, Tech. Rep., 2009.
2. R. L. Peter Lucas, Joseph Ballay, “The wrong cloud,” MAYA Design, Inc, Tech. Rep., 2009. [Online]. Available: http://www.maya.com/filedownload/126/The%20Wrong%20Cloud.pdf
3. M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “Above the Clouds: A Berkeley view of Cloud Computing,” *University of California, Berkeley*, 2009. [Online]. Available: http://d1smfj0g31qzek.cloudfront.net/abovetheclouds.pdf
4. J. Hayes, “Cred - or croak?” IET Knowledge Network, Tech. Rep., 2008. [Online]. Available: http://kn.theiet.org/magazine/issues/0820/cred-croak-0820.cfm?SaveToPDF

5. P. Mckenna, “Can we stop the internet destroying our planet?” *New Scientist*, vol. 197, no. 2637, pp. 20–21, 2008.
6. J. Kaplan, W. Forrest, and N. Kindler, “Revolutionizing data center energy efficiency,” McKinsey & Company, Tech. Rep., 2008. [Online]. Available: http://www.mckinsey.com/clientservice/bto/pointofview/pdf/RevolutionizingDataCenterEfficiency.pdf
7. Leiner, V. Cerf, D. Clark, R. Kahn, L. Kleinrock, D. Lynch, J. Postel, L. Roberts, and S. Wolff, “A brief history of the internet,” Institute for Information Systems and Computer Media, Tech. Rep., 2001. [Online]. Available: http://www.iicm.tugraz.at/thesis/cguetldiss/literatur/Kapitel02/References/Leineretal.2000/brief.html?timestamp=1197467969844
8. J. Scanlon and B. Wieners, “The internet cloud,” The Industry Standard, Tech. Rep., 1999. [Online]. Available: http://www.thestandard.com/article/0,1902,5466,00.html
9. Foster, Y. Zhao, I. Raicu, and S. Lu, “Cloud Computing and Grid Computing 360-degree compared,” in *Grid Computing Environments Workshop*, 2008, pp. 1–10.
10. T. Foremski, “Sun services CTO says utility computing acceptance is slow going,” ZDNet, CBS Interactive, Tech. Rep., 2006. [Online]. Available: http://blogs.zdnet.com/Foremski/?p=33