Technical Report

UCAM-CL-TR-863
ISSN 1476-2986

Number 863

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

Regional clouds: technical considerations

Jatinder Singh, Jean Bacon, Jon Crowcroft,
Anil Madhavapeddy, Thomas Pasquier,
W. Kuan Hon, Christopher Millard

November 2014

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Regional Clouds: Technical Considerations*

Jatinder Singh+, Jean Bacon, Jon Crowcroft, Anil Madhavapeddy, Thomas Pasquier
Computer Laboratory, University of Cambridge

W Kuan Hon and Christopher Millard
Centre for Commercial Law Studies, Queen Mary University of London

1. Introduction

The emergence and rapid uptake of cloud computing services raise a number of legal challenges. At the core of these lie issues of data management and control: where data can flow, who has potential access, and under what circumstances. Recently, there have been calls for *regional clouds*; where policy makers from various states have proposed cloud computing services that are restricted to serving (only) their particular geographic region. An example is the proposal for a Europe-only cloud.[1] Though there is often little detail surrounding the rhetoric – indeed, the concept is fraught with questions and complexity[2] – it generally represents an attempt at greater governance and control. The motivations are various, for example, to provide some certainty over the applicable legal regime(s), the ability to facilitate and/or hinder (particularly foreign) law-enforcement/governmental access, to bring about competitive advantage, and so forth.

This paper explores the technical considerations underpinning regional clouds, including the current state of cloud provisioning, what can be achieved using existing technologies, and the potential of ongoing research. For ease of explanation, we use a hypothetical Europe-only cloud as an example. Far from advocating rigid balkanisation,[3] we rather feel that from the technical perspective the concerns are rooted in the mechanisms for control.[4] Thus, assuming that the policy surrounding the various concerns can be specified precisely, in this paper we consider the technological implications of two issues: how can compliance with such policy be enforced and demonstrated.

Our discussion covers technology at various system levels, including network-centric controls, cloud platform management (hypervisors, virtual machines, containers), and governance mechanisms for providers, applications, tenants (i.e. customers who contract with cloud providers) and end-users. Note that in this paper, we use the term "users" to refer collectively to tenants and end-users (e.g. employees or customers of tenants) of a cloud service.

1.1 Cloud computing

Advances in networking, bandwidth, resource management and virtualisation technologies have resulted in service models that involve provisioning computing as a service. *Cloud computing* as it is called involves cloud *providers,* those offering the service, provisioning and managing a set of technical resources, amongst *tenants*: those consuming the cloud services through direct relationships with providers. The provider's business model is able to leverage economies of scale by sharing resources across tenants, while tenants gain from being able to pay for the resources they require, thus removing a costly start-up base, being able to acquire service *elasticity*—to rapidly scale-up and/or scale down resources in response

* With thanks to David Eyers, Carlos Molina-Jimenez, Divya Muthukumaran and Peter Pietzuch for their assistance. This paper was written for, and takes into account feedback from, the Microsoft Cloud Computing Research Centre (MCCRC) Symposium, September 2014: "A Cloud for Europe? Feasibility and Implications of Internet Regionalisation". Both the research presented in this paper, and the symposium at which it was discussed, were made possible as a result of generous support from Microsoft. The views presented herein are, however, the authors' alone.
N 1 represents a complementary paper, focusing on the legal, policy and regulatory aspects of the topic.

+ Contact: jatinder.singh@cl.cam.ac.uk

[1] W Kuan Hon, Christopher Millard, Chris Reed, Jatinder Singh, Ian Walden and Jon Crowcroft . *Policy, Legal and Regulatory Implications of a Europe-only Cloud, Discussion Paper.* (2014). http://ssrn.com/abstract=2527951

[2] N 1.

[3] Indeed, this was the consensus reached at the MCCRC Symposium.

[4] For a detailed analysis of the legal, policy and regulatory concerns, see n 1.

to demand fluctuations—and more generally improving access to storage and computational services. The end-users of a system may interact with a cloud provider either directly or indirectly: some may use the services that the tenant provides, or interact directly with the provider, depending on the service offered. Providers may use other providers' services, as *sub-services*, in order to provision their own offering. As such, there are a number of stakeholders simultaneously involved in providing the computing services, with the legal complications that ensue—even more so if different countries claim jurisdiction to regulate all or some of these services or sub-services.

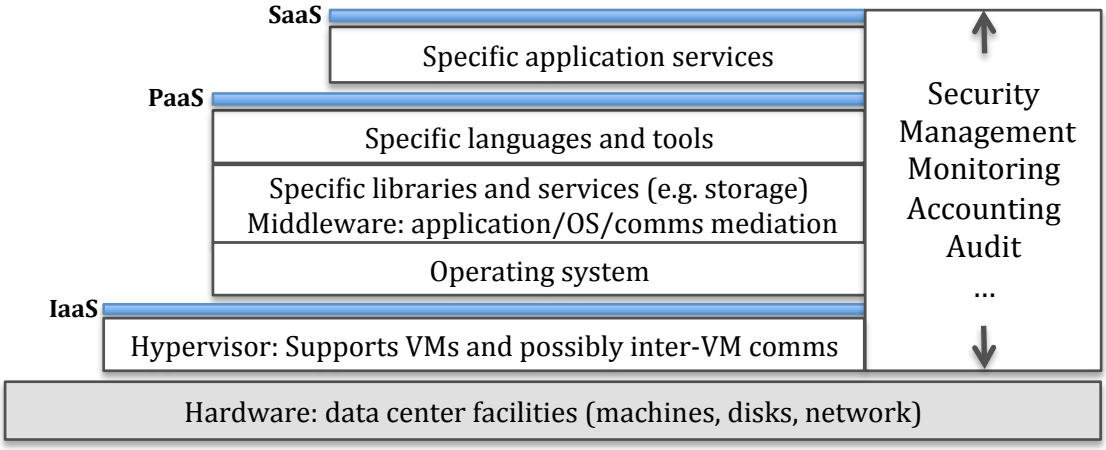

Figure 1: Cloud service levels and their associated offerings (Generally, at each service level, providers offer all below the blue line, tenants manage all above)

Figure 1 shows that cloud services are provisioned at different levels, the most common service models being:

Infrastructure as a Service (IaaS): A common cloud provisioning approach involves the use of *virtual machines (VMs)*, which from the tenants' perspective in many ways resemble a physical (hardware) server. Tenants install an operating system (often from an *image*, encapsulating a preconfigured system), and are responsible for installing and managing their cloud applications and other software running on their installed operating system (OS). Providers manage VMs through the *hypervisor*, which operates across tenant VMs, and regulates their interactions with the provider's hardware. The provider may isolate and/or share physical resources (I/O) across VMs, which can belong to different tenants. Just how much is shared depends on the service offering; e.g. Amazon Dedicated Instances[5] provides tenant-dedicated hardware, and there is research on how to prevent competitors sharing physical resources.[6][7] End-users interact with the provider indirectly, through the software/services that the tenant provides. Example IaaS providers include Amazon EC2, Microsoft Azure, Google Compute Engine, to name a representative few.

An alternative to the hypervisor approach is *containers,* or *operating system virtualisation*, where tenants share the same OS, but are isolated (contained) from each other in terms of resource (CPU, memory, I/O) usage and allocation. Containers relate to both IaaS and PaaS (below), as the providers offer the OS, typically running directly on hardware, with varying levels of other services. Example OS-level virtualisation implementations include HP-UX, Amazon Docker, LXC and Parallels Virtuozzo.

Platform as a Service (PaaS): Represents a higher-level service offering, providing tenants with a framework (stack) for application/service development, with the provider hosting the application/service. This is to aid tenants in rapid application development and deployment, and simplified management—often with built-in scalability capabilities. Again, end-users interact with applications provided by the tenants. Examples include Microsoft Azure, Salesforce Heroku, AWS Elastic Beanstalk, etc.

---

[5] "Dedicated Instances - Amazon Virtual Private Cloud." 2012. 7 Aug. 2014 (http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/dedicated-instance.html)

[6] Wu, Ruoyu et al. "Information flow control in cloud computing." *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010 6th International Conference on* 9 Oct. 2010: 1-7.

[7] Guanhai Wang, Minglu Li, and Chuliang Weng. "Chinese wall isolation mechanism and its implementation on VMM." *Systems and Virtualization Management. Standards and the Cloud* (2010): 13-18.

Software as a Service (SaaS): Provides the entire end-user application, meaning that not only is the implementation and deployment provider-managed, but the service includes the application functionality itself. End-users directly engage with the provider's applications; think Facebook, Google Mail, Office 365. *Tenants* may lease the service—perhaps rebranding the applications(s) for end-users, such as tailored shopping carts, or a University using a webmail offering to provide institutional email—however the software assets are typically controlled by the SaaS provider. The degree of isolation between tenants will depend on the service offering; some tenants might be given dedicated application and storage infrastructure, in other cases all or part of the infrastructure might be shared amongst tenants.

The different cloud service models enable different degrees of management and control on the part of tenants. The technology aiming at governance and compliance must be appropriate to the scope of the service offering.

1.2 Trust in providers

A cloud provider offers services, be they applications, computational services, storage, infrastructure, etc., involving end-user and tenant data. Thus, the very nature of cloud services entails a degree of *trust* between end-users, tenants, and providers.

As explored in *Cloud Computing Law*,[8] in many cases provider contracts are providers' own standard terms, where tenants often have little room for negotiation: "take it or leave it". This either boxes potential tenants into agreeing particular terms (whether fully comprehended, or not), or may be a factor in electing not to use the service.

The result is that with public cloud services, tenants and end-users generally have little input over how the service is actually provisioned, with the provider retaining the power and flexibility to manage such details as it sees fit. As such, the specifics of service provision are generally kept from users; the cloud service being opaque, with the ability for users to monitor and audit the internals of service provision limited.

This, implicitly, entails a great deal of trust that users must place in a provider to appropriately manage their data, at least to provide a level of protection as agreed in the service contract.[9] Further, a provider will often enlist the services of others to enable provision, e.g. network providers, software/storage services, other cloud providers, etc. Again, there are often limited, if any, mechanisms for user oversight. These issues are of particular concern given that a user may bear responsibility for data, e.g. by way of data protection obligations.[10] Perhaps some rely on the economic and reputational consequences for a provider in the case of a significant (noticeable) failure; going with a provider "big enough to sue". Such a mentality, of course, favours the larger, more well-known providers.

The level of trust a tenant must place in its provider relates to the service model. For IaaS systems, the tenant has some direct control over the codebase, operating system and software setup, but it must trust that the provider ensures proper isolation between VMs, to protect against interference by, and accidental data disclosures, to other tenants. Tenant control diminishes as we move up the service-stack of provider offerings (Fig. 1).[11]

Underpinning this is that a cloud user must also trust that a provider will have in place the appropriate arrangements with any *sub-providers*—collaborative entities that assist in service provision[12]—to ensure proper handling of their data. Further, they must also trust that the provider(s) will not improperly interfere (or 'snoop') on their virtual machines and/or stored data. In this respect, the loss of control becomes even more evident as we move up the provisioning stack. For instance, IaaS tenants manage

[8] Christopher Millard (ed.). *Cloud Computing Law*. Oxford University Press, 2013.

[9] In this paper we refer to *trust* as appropriate to data management concerns; though also relevant, we do not consider aspects of service provisioning, such as performance guarantees.

[10] As discussed in *Cloud Computing Law* chapter 8, there are also legal issues of how such liability extends to *sub-providers*: collaborative entities that assist in service provision.

[11] And see *Cloud Computing Law* chapter 2 section 5.

[12] Services may be provided at similar or different-layers of the technical stack to that of the consumer-facing cloud provider. For instance, Dropbox uses Amazon S3 to provide its (non-metadata) storage capability, though user interactions are directly with Dropbox.

their operating system and applications, and thus have more flexibility regarding data management and processing, but in SaaS the applications are already defined leaving few, if any, mechanisms for control. Indeed, many SaaS providers make it explicit that they will use uploaded data for various purposes such as advertising—Facebook and Gmail are cases in point.

A focus of ongoing technical research is to design and develop technical mechanisms for improving trust in cloud services, through mechanisms enabling more control for cloud users. We explore this work later in this paper.

1.3 Jurisdictional concerns

There is a strong argument that the regulatory concerns affecting cloud computing are better driven by considering who has access to intelligible data and which countries have jurisdiction over such persons, rather than the physical location of service offerings *per se* being the main determinant.[13] However, conflicts of laws are a perennial problem, with cloud as with the Internet generally. One issue, for example, is the fact that there may be multiple jurisdictions applicable to a tenant, provider, and in turn any sub-providers, for a particular service scenario, and countries are increasingly attempting to apply their laws extraterritorially. Which jurisdictions' laws should prevail, when multiple jurisdictions may be relevant? That question cannot be solved by technology, but requires international political agreement.

While it is clear that some aspects of data management policy are independent of the technical concerns, such as some legal, economic and social concerns, others are very much in line. For instance, it is argued that a sensible approach to governance is one that takes into consideration both a) providers' capability to access user data and b) jurisdictions under which they operate.

The former capability refers to the provider's access (or ability thereof) to intelligible user data, e.g. a storage service holding encrypted data but with no access to the keys is arguably less of a concern than one that holds the decryption keys and/or operates on data in the clear. Here, technical mechanisms can help manage this capability.

"Jurisdiction" need not refer to physical location, but rather one can imagine a scenario where providers may offer services that comply with minimum requirements as stipulated by the laws of particular jurisdictions, regardless of the physical location of the infrastructure used or of the jurisdictions of the provider or any sub-providers. We term this *virtual jurisdiction*, and mention this in a technical context in §5.3 and §7.3. Indeed, other concerns may also be relevant, such as the jurisdiction of the provider's place of incorporation. Again, technical mechanisms can assist in managing data in accordance with such concerns, such as only allowing information flows within a particular "virtual jurisdiction", assuming the constraints can be defined and higher-level issues resolved.

1.4 Focus of discussion

The issues described above drive the technical discussion of this paper. To be clear, our focus is on providing technical measures to improve compliance and assurance in the cloud in accordance with any relevant contracts, agreements and national laws. We do not directly consider protecting against a malicious agent and/or surreptitious actions by government agency—these present a different set of technical considerations.

We now explore work from a number of technical perspectives towards control and assurance mechanisms, particularly those that operate beyond the interface of provider and the tenant, extending into provider (and sub-provider) territory.

2. Communications and Localisation

The enforcement of laws affecting cloud computing is a multidimensional problem. In this section we focus our attention on communication-related issues and discuss broader concerns before introducing some ideas about storage (§3) and computation (§4), which are also relevant to the enforcement of regulations affecting cloud data.

With respect to communication paths, there are three main considerations:

[13] See n 1 for detailed discussion of the issues described in this section.

Path transparency: This considers the means for determining, *a priori*, the properties of a particular network path. For example, where the physical links and switches are, who runs them, what software they run, etc.?

Path controls: This concerns control over connections. There are, for example, mechanisms for determining your network provider. *Multihoming*, being connected to more than one network, is possible, as is particularly evident with mobile devices. Similarly, it is common that servers/data centres are serviced by a number of networks. *Virtual private networks* (VPNs) provide another mechanism to direct flows through particular communications infrastructure. Internet service providers (ISPs) have the power to control paths directly, though it is the ISP that defines the policy, not its customers.

At the protocol level, *IPv4* and *IPv6 loose source routing* enables one to define aspects of a routing path, but this option is normally not available as network providers prefer to manage routing themselves, and because it can open the possibility for indirect denial of service attacks by bypassing firewalls, etc. There is also research into routing mechanisms that take a more data-centric rather than address/path-centric approach, such as those based on rendezvous and publish/subscribe.[14]

Path monitoring: Is it possible to determine that your data flows in accordance with the declared path properties? That is, is your data transferred in the ways you expect? Here, regulators and monitors,[15] along with technical tools such as `traceroute`, `geolocation`, `ping`, etc., can assist.

2.1 Localisation

The topic of localisation is relevant to the enforcement of regulations affecting cloud computing. One can argue that the cloud is nothing but a business model for using computing resources. Yet as soon as some code is executed and some data are stored, then there is a mapping of that program, the data and other resources involved, to a physical location. This mapping is not trivial to understand and examine—in particular when the parties involved are mobile— without the appropriate techniques. In line with this, the theoretical work on bigraphs conducted by Professor Robin Milner (University of Cambridge) and others might help as a modelling tool. A bigraph is a rigorous generic model that can be used for modelling the concurrent and interactive behaviour of populations of mobile communicating agents. Through their graphical presentation, bigraphs can make it easy for non–experts to visualise their system and assemble them geometrically, which can include physical locations.

2.2 Topology, topography, network administration

The notions of *topology*, the arrangement of elements of a network, and *topography*, the geographic location of these elements, have long been considered with respect to network management. We now discuss these notions and some related practical tools.

1. Geo–location services: It is possible to place a user within a geographic location based on their IP address. Although the location can be imprecise, e.g. perhaps the wrong town/city, the way that IP-addresses are allocated means the user's country is fairly certain.[16] As such, it is routine nowadays for services to use geo-location to filter or adapt their content.[17] For example, the BBC does this to stop end-users outside the UK accessing iPlayer content (and it also does it to stop end-users inside the UK from seeing commercial BBC world content). The same technique is used for geo-located targeted advertising. YouTube, for example, uses address-based geo-location to control what music and videos it delivers depending on whether it has negotiated rights to do so (e.g. in exchange for advertising revenue or analytics for the rights owner) in a given geographical region. This implies a mapping from the geographic location of the browser/computer/client, to a geographic region for the purposes of intellectual property ownership/licensing. It is clear that the same approach can easily encapsulate other location-based concerns, such as a data management regime relevant to tenants operating in a particular region.

---

[14] For examples, see http://i3.cs.berkeley.edu/ and http://www.psirp.org.

[15] For example, https://www.samknows.com/broadband/about provides data about broadband performance.

[16] http://whatismyipaddress.com/geolocation-accuracy

[17] Though geolocating an IP address does not guarantee a client's geographical location, which may be obscured by corporate networks or VPNs, in practice it is generally considered sufficient for the purposes of content filtering, given the majority of users do not use such services.

2. XenoSearch: When the Xen hypervisor[18] was first designed, tools were also developed for end-users to launch VMs in specific locations.[19] Thus, it was perfectly feasible to start a virtual machine in a *specific* data centre. The same applies to storage. Leading cloud providers like Amazon offer location–aware instantiation of EC2 (compute) and S3 (storage). Early tools were more aimed at solving constraints, e.g. managing latency,[20] but tools could target other purposes. Cloud providers know where their data centres are located (geographically, administratively—in terms of maintenance and support—and for the purposes of billing, etc.) so this is trivial compared with the geo-location services for (potentially mobile) browsers.

There are more nebulous services (like Gmail and search) which run over large distributed infrastructures, which may themselves run over even larger lower layer infrastructures. In fact, both Gmail and YouTube have large distributed storage systems that may not currently map well to a location, but the argument above says that they easily could.

3. Border Gateway Protocol: In the early Internet days, around 1992, a system called the Inter–domain routing system was devised, which uses a protocol fittingly called the *Border Gateway Protocol (BGP)*. BGP, which is now in widespread use, is a mechanism for controlling the flow of traffic between regions of the Internet called *Autonomous Systems (ASs).*[21] While these regions are topological in nature, rather than topographical, as in point 2, all Border Routers are physically, statically located in or near *Internet Exchange Points (IXP)* (point of traffic exchange between ASs). Currently many ISPs are also operators in the 'telco' sense, within a specific country, and so their infrastructure in terms of AS topography (and, likely, legal boundaries) is well defined. This would be obviously true within the UK, France, Spain, Italy, Germany, etc, where the national largest ISP is also the telco, i.e. BT, Orange (formerly France Telecom), Telefonica, Telecom Italia, Deutsche Telekom, etc. In these situations it would be fairly easy to map information flow at the network level and constrain it by means of routing policies. For example, BGP has rules for traffic ingress, egress and transit that can be applied on a per IP–prefix basis, giving controls over routing to/from sub-networks. So if necessary, one could constrain traffic from a given cloud provider using today's existing network technology. BGP is capable of capturing a lot of complex network business relationships and controlling data flows accordingly, and might be a useful source of approaches to constraining where cloud data may and may not go.

4. Traffic localisation: Some telcos/ISPs/ASs have also integrated the content delivery infrastructures with their backbone networks to control where the traffic goes, to avoid unnecessary transit fee costs from other ISPs. For example, Telefonica in Spain works directly with content/application delivery provider Akamai to optimise traffic flow from streaming and Web content servers to stay within their physical network within Spain specifically, although they obviously have many more networks–most of Latin America. Localisation of traffic is good for the provider, in terms of cost, and users, for reasons of reducing latency. Thus, requiring such controls is not a great burden, but more aligned with the network providers' business interests. In line with this, work such as P4P[22] aims at reconciling conflicts between P2P (peer to peer) networks and ISPs, by allowing the applications to be involved in lower-level routing processes.

5. Internet is becoming "flatter": The model of how the Internet fits together in terms of IXPs, i.e. exchanges between major networks, is always changing. This is not only in line with general routing requirements, but also due to business/peering arrangements. It is said the Internet is becoming "flatter",[23] in the sense that more and more ASs strategically peer at specific IXPs to enable more efficient traffic exchanges by avoiding higher tier providers. From a cloud perspective, many cloud providers

---

[18] Barham, Paul R et al. "Xen 2002." *University of Cambridge, Computer Laboratory, Tech. Rep. UCAM-CL-TR-553* (2003).

[19] Spence,D.,Harris,T.:XenoSearch: Distributed resource discovery in the XenoServer open platform. In: Proc. 12th IEEE Int'l Symposium on High Performance Distributed Computing (HPDC'03).

[20] In this case, reducing network overheads (thereby increasing performance and speed for users) by physically locating the VMs close to the services it leverages and users it services.

[21] A part of the network (thus, a set of routers) under clear single administrative control (often that of telcos, ISPs and very large companies), that operates a defined routing policy.

[22] http://www.cs.yale.edu/homes/yong/p4p.html

[23] Ager, B., Chatzis, N., Feldmann, A., Sarrar, N., Uhlig, S., Willinger, W.: Anatomy of a large European IXP. In: Proc. ACM SIGCOMM'12. (2012).

connect their regional data centres to the major infrastructure service providers in an area for cost and performance gains. In line with this, enforcing information flow controls (see §7) at the network layer would likely involve Amazon, Google, Microsoft, or Facebook.

2.3 Broader discussion

For Europe, the continental ISPs may be concerned about routing their data through the UK, because of the GCHQ intercept[24] and *Five-Eyes*[25] sharing arrangements effectively rendering data that traverses any link physically located in the UK potentially accessible to US authorities too. The same does not appear to be true if data is routed between (say) France and Italy via Switzerland.

The implication is that if you are just routing end–to–end encrypted data, then the risks are relatively small. Therefore the recent (post-Snowden) practices of routing data between data centres in encrypted form helps address this.[26] However, tenants may move data within and between services; thus data may be stored in places at risk, since keys might be accessible to cloud providers, data may be moved by applications, or stored unencrypted. This issue needs to be regarded as a sort of hierarchy of risks.

Other questions that one can ask include whether data travels via switches or routers in other countries. If so, is the information encrypted with keys stored and signed with certificate authorities that are *not* subject to the jurisdiction of those other countries? Might cloud data be stored on servers physically located in other countries? If so, are the data stored encrypted, with keys for decryption only kept elsewhere? Overall, the concern here is about where the data are encrypted and decrypted, given their transfer over different geographical (for example, countries) and organisational boundaries (for example cloud providers, ISPs). That is, assuming one may access encrypted data, the pertinent question is: who can access and who manages (creates, certifies, distributes, revokes, backs-up, escrows, etc.) the encryption and decryption keys?

BGP allows control of traffic flow, for example so as not to ingress/egress or transit a given AS, which represents a portion of network under particular single control. In this manner, if we know that any element of an AS may be subject to the jurisdiction of an untrusted country, (or just physically located somewhere we do not trust in general), then the external BGP routers (eBGP) may be configured by network providers to prevent them from routing traffic to the untrusted AS. At the technical level, the configuration involves programmatic manipulation of the routing policies that dictate the operation of the eBGP routers involved, and can be done by the router managers.[27] At a business level, this implies collaboration between cloud providers, telcos, ISPs and ASs, (see point 4 of §2.2 Traffic localisation). Generally, routing tends to follow business relationships, provider or peering arrangements for direct data exchange between network service providers, as configured via contracts.

One limitation to take into account is that is that any such eBGP configuration is “all or nothing”, meaning that it applies to all traffic, regardless of what the data actually represents. There is also still the need for management at higher-levels (storage, caching, location of keys, certificate authorities, etc.)

A more general question that arises here is what is the motivation for partitioning a network according to jurisdictional concerns?
It would seem:

1. For compliant cloud providers to be able to advertise that their services do not allow data flow beyond the relevant borders (transparently) and,
2. To be able to measure the extent to which a "miscreant" cloud provider has allowed data or computation to flow (or data to be stored) where it should not (and produce evidence to that effect that will stand up in court).

[24] http://www.theguardian.com/world/the-nsa-files

[25] https://www.nsa.gov/public_info/press_room/2010/ukusa.shtml

[26] http://gmailblog.blogspot.co.uk/2014/03/staying-at-forefront-of-email-security.html

[27] Basic Configuration Examples for BGP, Juniper Networks 2001, http://jncie.files.wordpress.com/2008/09/350008_-basic-configuration-examples-for-bgp.pdf

3. Data storage

The issue here is to provide users (individuals or institutions) who use cloud computing to store data with assurance, and ideally means for verification, that their data are stored where they wish (for example, not outside the EEA), so that their processing complies with applicable laws or regulations. Also users need assurance that the data are retained and available (in readable format) to the entitled parties in accordance with their own expectations and laws and regulations with which they must comply.

Equally important, customers need to be assured that their data are deleted when any relevant retention period expires. For example, EPSRC Policy Framework on Research Data dictates that "Research organisations will ensure that EPSRC–funded research data is securely preserved for a minimum of 10-years from the date that any researcher privileged access period expires or, if others have accessed the data, from last date on which access to the data was requested by a third party."[28]

It is worth mentioning that some cloud providers offer their tenants technical means of selecting the physical location (for example, US, Europe, Asia Pacific, etc.) of their resources. With Amazon for example, a tenant is able to dictate that his S3 storage is located in the US, Europe, Asia Pacific, etc. However constraining stored data to particular geographical locations is not the solution in and of itself.[29] The challenge that cloud computing presents is that due to lack of transparency, constraining physical location of data to a particular geographical region does not guarantee security. Further, users in general are not necessarily aware of the existence and location of cached and/or backup copies of their data.

4. Computation

When cloud computing services are used to perform computing operations on data, the main challenge is to provide users with assurance that the software that accesses their data is doing only what it is supposed to do, instead of accidentally or deliberately leaking sensitive information, and that other software is not accessing the data or monitoring the user's operations on data. This is still an open research problem, though it is being actively addressed (see §5.1 and §7.3). With current practices, users have to blindly trust the provider's service stack (Fig 1)—and more generally, the provider itself, given that computation generally happens on intelligible data (see §6.2), so that generally even encrypted data would have to be decrypted to enable computation. Ideally, users (or attestation services) should be able to examine the provenance of the software and verify records about its origin, maintenance, testing, etc. In addition, there should be the means to control and verify information flows to and from various software services.

5. Provisioning the cloud

The economics of the cloud depend on increasing the usage of physical computers by intelligently operating services for multiple users across them, taking advantage of the statistical property that not all customers will require full utilisation of the physical compute resources at any given time. A fundamental guarantee required from the cloud infrastructure is *multi-tenant isolation*, ensuring that the actions of one party are isolated from the other (usually untrusted) parties who happen to be running operations using the same physical hardware and/or software (for PaaS/SaaS).

For IaaS deployments, this isolation guarantee is provided by the *hypervisor*, which is software that controls the physical hardware and manages the lifecycle of *virtual machines* that run under the illusion that each one is a complete standalone operating system installation. The strongest isolation guarantees are provided by so-called Type 1 hypervisors such as Xen, Hyper-V and VMware that are the first piece of software that boots on a physical host.

There are several layers at which the software stack can be improved in terms of trustworthiness, explored in the following sections.

5.1 Trusted computing base disaggregation

Existing hypervisors depend on a privileged "domain-0" virtual machine that has access to physical hardware. This VM runs device drivers from Windows (Hyper-V) or Linux (Xen, VMware) and proxies

[28] http://www.epsrc.ac.uk/about/standards/researchdata/

[29] For example, see https://www.scl.org/site.aspx?i=ed35439

traffic between the unprivileged application VMs of tenants. Any attacker that can gain access to the domain-0 virtual machine effectively controls all of the other VM resources, so guaranteeing isolation at this layer is vital to enforcing any other security properties throughout the system.

In the Xen hypervisor, recent versions support the notion of *driver domains* that break up the monolithic domain-0 into multiple, specialized VMs that run a small portion of the physical device space (such as just the network card, or the storage array). Thus, if one privileged VM crashes, only that physical device is compromised, thus compartmentalizing attacks.[30]

There is also increasing support within the hardware for enforcing efficient isolation. Intel and AMD CPUs have support for nested virtualisation that not only permits hypervisors to run recursively (enabling virtual environments within virtual environments), but also improves performance isolation by protecting some system resources (such as the TLB cache) across context switches (where the CPU switches between performing different computational tasks). Network and storage devices can be multiplexed across (i.e. have their use shared between) VMs more flexibly, with multiqueue support and hardware passthrough to avoid or reduce performance overheads. Recent work has leveraged these new hardware capabilities to protect (unmodified) applications from an untrusted cloud host.[31]

5.2 Reducing legacy code exposure

A key driver of the growth of cloud computing in the early days was server consolidation. Existing applications were often installed on physical hosts (servers) that were individually underutilised, and virtualisation made it feasible to pack them onto fewer hosts without requiring any modifications or code recompilation. While operating-system virtualisation (containers) is undeniably useful, it adds yet another layer to an already highly-layered software stack now including: support for old physical protocols (e.g., disk standards developed in the 1980s, such as IDE); irrelevant optimizations (e.g., disk elevator algorithms on SSD drives); backward-compatible interfaces (e.g., POSIX); user-space processes and threads (in addition to VMs on a hypervisor); and managed-code runtimes (e.g., OCaml, .NET, or Java). All of these layers sit beneath the actual application code that is executing the business logic at hand.

These software layers are not just inefficient: they present a real threat to the trustworthiness of a system by adding complexity and attack surfaces via software bugs. One solution to this arises from the insight that services deployed to the cloud are normally highly specialised (i.e. a web server, or a database, or an analytics engine), and assembled for that purpose via coordination engines such as Chef or Puppet. However, the deployed images are rarely optimized to remove the unnecessary portions.

One recently-developed technique[32] explores the benefits of automating such specialization. *Unikernels*[33] are specialised virtual machine images compiled from the full stack of application code, system libraries and configuration. The resulting images are often orders of magnitude smaller in terms of image size, due to the elimination of unnecessary features when the image is built.

From a security perspective, unikernels combine many of the advantages of the container- and hypervisor-based approaches.[34] For instance, it is possible to build even more trustworthy unikernels by adopting a safer programming language to build the source code for the entire appliance. The Mirage[35] project from Cambridge uses the statically-typed OCaml programming language, and the HalVM by Galois uses Haskell. Both these unikernel systems build not only the application logic in a (type and memory)

---

[30] Patrick Colp et al. "Breaking up is hard to do: security and functionality in a commodity hypervisor." *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles* 23 Oct. 2011: 189-202.

[31] Andrew Baumann, Marcus Peinado, and Galen Hunt. "Shielding applications from an untrusted cloud with Haven." *Proceedings of the 11th USENIX conference on Operating Systems Design and Implementation* 6 Oct. 2014: 267-283.

[32] Madhavapeddy, Anil et al. "Turning down the LAMP: software specialisation for the cloud." *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing, HotCloud* 22 Jun. 2010: 11-11.

[33] Madhavapeddy, Anil et al. "Unikernels: Library operating systems for the cloud." *ACM SIGPLAN Notices* 16 Mar. 2013: 461-472.

[34] https://www.linux.com/news/enterprise/cloud-computing/785769-containers-vs-hypervisors-the-battle-has-just-begun/

[35] http://www.openmirage.org

safe style,[36] but also the system libraries that are conventionally written in C, such as the TCP/IP network stack or device drivers. This is normally impractical for real-world deployments due to the sheer diversity of physical devices, but is possible when running in a cloud environment, as the virtual hardware environment exposed by the hypervisor is simple and abstracted away from the actual physical devices that underpin it.

Unikernels also make it possible to track much more efficiently the source code *provenance* of components that go into a deployed system. It can be very difficult to keep precise track of what goes into a typical Linux or Windows distribution due to the loose coupling of components. Since unikernels are mechanically assembled from *all* relevant libraries (including OS, application and configuration), a full view of the source code can be stored and tracked for security issues.[37]

5.3 Mobile and personal clouds

The "cloud" currently consists of services offered by a set of providers, whose services are leveraged by an ever-increasing number of mobile devices, such as smartphones, tablets, etc. The commercial reality mirrors the 'thin-client' approach of yesteryear: a 'centralised' service offering accessed by 'dumb terminals'. This service model tends to overlook the powerful compute capabilities of the devices.

Many in academia are working on the opposite extreme: building distributed systems, including mobile and decentralised peer-to-peer networks, distributed file systems, ad-hoc mesh networks, etc. We have termed such systems *The Mist,*[38] as these aim at dispersing data among several responsible entities (providers *and other users*), rather than relying on a single monolithic provider.

Comparing the two approaches, which represent the extremes, we have proposed *droplets* as a controlled trade-off.[39] Made possible by unikernels, a droplet encapsulates a unit of network-connected computation and storage that is designed to migrate around the Internet (service providers) and personal devices. In this way, one can imagine droplets enabling *personal clouds,* encapsulating specific services and functionality for individuals, that can be made available and processed locally and/or by other users or service providers, depending on the requirements, context, incentives and risks at the time.

If the world moves this way, a number of interesting challenges are raised, concerning what constitutes a provider (potentially anyone), ownership of rights relating to data (particularly for shared services), the laws that should apply, and the mechanisms for managing this.

More generally, because unikernels are small and encapsulate specific functionality (in accordance with a build policy) they can be designed, migrated and deployed in as appropriate to regional considerations. That is, it is possible to envisage particular unikernel instances that target different virtual jurisdictions (see §1.3 and §7.2.4).

6. Restricting access

Those who have rights and/or obligations in relation to data have an explicit interest in defining the terms under which data may be accessed. Not all data are equal; rather, different levels of protection and governance will be required depending on the data, and the associated obligations, risks and other circumstances.

We now describe two categories of methods for governing access to data: authentication- and authorisation-based access controls, and encryption. The former refers to allowing a *principal*[40] to

---

[36] Memory safety ensures that bugs in the software logic cannot result in arbitrary code execution through buffer overflows (https://www.owasp.org/index.php/Buffer_Overflow). Static type safety permits higher level logical invariants to be enforced at compilation time, rejecting applications that violate them before the unikernel is ever executed.

[37] Anil Madhavapeddy, and David J Scott. "Unikernels: the rise of the virtual library operating system." *Communications of the ACM* 57.1 (2014): 61-69.

[38] Jon Crowcroft et al. "Unclouded vision." *Distributed Computing and Networking* (2011): 29-40.

[39] Ibid.

[40] A *principal* is an entity operating within a security context. Thus, depending on the situation, the term can refer to human users, applications, software, threads, connections, roles, and so forth.

perform some action in accordance with authentication and authorisation rules, while encryption mechanisms work to scramble the data, making it generally incomprehensible to anyone except for those holding the requisite cryptographic keys.

6.1 Authentication- and authorisation-based access controls

This form of access control aims at governing the actions that may be taken on objects, be they accessing particular data (a file, record, data stream), issuing a query, accessing a resource, performing some computation, and so forth. The controls are typically principal-focused, in the sense that control policy governing a particular action is defined to regulate those undertaking the action, enforced when they attempt to take that action.

It follows that there are two main aspects to such controls:

Authentication concerns proof of identity. That is, determining who the principal is: are they who they say they are? Familiar forms of authentication include login/password, smart cards and biometric systems. Once authenticated, a principal may be assigned various credentials to assist other authentication processes, and in making authorisation decisions.

Authorisation entails determining whether a principal is allowed to undertake an action that they wish to perform. Such governance is defined in an access control policy that, depending on the model, may encapsulate a variety of concerns, including attributes of the principal ('who' they are, their credentials, how they were authenticated), the action, the object(s) involved, and in some cases the environmental context, e.g. time, location of the principal, etc. This policy tends to be evaluated at the time a principal attempts (or requests) to take the action.

Access control technology has a long-standing history. Some of the earlier forms fall into the category of *Mandatory Access Controls* (MAC), where there is a globally-defined set of security labels applied to data to determine levels of access to that data—e.g. military-esque classification systems `secret`, `top secret`, where only those with particular clearance can access objects marked as such. *Discretionary Access Controls* (DAC) give more flexibility, in that policies are tailored with respect to individual objects/actions. An example of DAC is that of permissions in most operating systems' file systems, wherein users may specify the (`read/write/execute`) privileges over particular files.

With respect to policy, Access Control Lists are a common mechanism that encodes policy in a DAC manner by associating a list of principals that are permitted to perform particular actions. Role Based Access Control (RBAC) is a paradigm that aims to simplify management, by associating principals with roles, and roles with privileges. This allows, for example, simply adding "warden" as a privilege for prisoner files, rather than having permissions being enumerated for each warden, individually. We have worked on more advanced systems that can include parameterised roles and environmental context,[41] in order to implement more flexible and manageable access control policy, e.g. where parameterised roles help prevent role-explosion by adding a contextual element to a role definition. Similar concepts have been applied to managing data flows within a publish/subscribe messaging middleware.[42]

For these controls, policy is enforced at the time of action, considering the principals directly involved in the interaction. In a cloud context, this means that the controls govern the cloud end-user-provider interactions at the interface between them. These mechanisms typically do not, by themselves, offer users control beyond that point, e.g. how their data is managed internally by the provider(s)'.[43]

6.2 Encryption

Encryption is the process by which data are encoded, according to some algorithm (or cipher), such that they are not intelligible or useful except for those able to decrypt the data. A key is used to specify the data transformation, such that those who have the decryption key are able to decrypt the data. Symmetric key

[41] Jean Bacon, Ken Moody, and Walt Yao. "A model of OASIS role-based access control and its support for active security." *ACM Transactions on Information and System Security (TISSEC)* 5.4 (2002): 492-540.

[42] Jatinder Singh, David M Eyers, and Jean Bacon. "Disclosure control in multi-domain publish/subscribe systems." *Proceedings of the 5th ACM international conference on Distributed Event-Based Systems* 11 Jul. 2011: 159-170.

[43] A provider will have its own access control systems, perhaps regulating actions with other sub-providers and as part of isolating tenants, though these predominantly encapsulate provider policy rather than consumer concerns.

cryptography defines schemes where the same key is used for both encryption and decryption, while asymmetric key cryptography schemes use a key-pair—two keys, mathematically linked, but where knowing one does not give away the other. Asymmetric key cryptography underpins public key cryptography, where, for example data encrypted using a public key can only be decrypted by the associated private key (and vice-versa). This forms the basis of digital signatures, as the keys can be used for authentication purposes.

Encryption provides an independent form of protection to the access controls described above. Encryption does not generally target physical access to data, but rather, data's usability. Access is regulated through the distribution of keys. In a cloud context, this means that if a user places encrypted data in the cloud, such data will not be accessible by the provider, or anyone else, unless they hold they requisite keys.

While encryption of data may be appropriate for storage services, and more generally to protect against more surreptitious attacks (snooping), many cloud service offerings involve computation. This means that generally a provider must have access to the customer's intelligible data (and/or their keys if data are encrypted) in order to provide the computation service.

There are movements towards addressing this. The concept of *encrypted search*[44] is one where criteria can be specified, returning the encrypted objects (data) that match. The data are stored, and remain, encrypted, and the search occurs over the objects in their encrypted form, and the search criteria are also protected and not revealed as part of the matching process. The expressiveness and capability of the search depends on the particular model.[45] Such functionality has been considered in the context of cloud computing, for example, to provide encrypted storage services.[46] (Though not encryption-based, it is worth here mentioning *differential privacy*[47], which regulates the queries on a dataset in order to balance the provision of useful, statistical-based results with the probability of identifying individual records.)

More powerful is the developing area of *homomorphic encryption*, which enables operations (transformations) on data in encrypted form. Research is moving towards practicable fully-homomorphic encryption, where any function can be computed on the encrypted data in a reasonable time. However this remains an ongoing research topic;[48] the state of the art entails a limited set of operations on the encrypted data and/or operations being too slow to be useful at scale.[49] Though anecdotally there seems to be some debate as to when such technology will become practicable for mainstream usage, there are movements in the right direction.[50] Indeed, fully homomorphic encryption offers much promise in the area of cloud computing, as it will enable providers to offer computational services where data remains encrypted.

One general issue with encryption is that key management is difficult.[51] Keys must be distributed to the relevant parties, and revoked (and reallocated) when conditions change. This might be due to a change in a principal's rights (e.g. an employee resigning) or perhaps due to a compromised key. Leveraging

---

[44] Dawn Xiaoding Singh, David Wagner, and Adrian Perrig. "Practical techniques for searches on encrypted data." *Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on* 2000: 44-55.

[45] Philippe Golle, Jessica Staddon, and Brent Waters. "Secure conjunctive keyword search over encrypted data." *Applied Cryptography and Network Security* 1 Jan. 2004: 31-45.

[46] Seny Kamara, and Kristin Lauter. "Cryptographic cloud storage." *Financial Cryptography and Data Security* (2010): 136-149.

[47] See Cynthia Dwork. "Differential privacy: A survey of results." *Theory and Applications of Models of Computation*. Springer Berlin Heidelberg, 2008. 1-19, and recently from Google: Erlingsson, Úlfar, Aleksandra Korolova, and Vasyl Pihur. "RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response." *arXiv preprint arXiv:1407.6981* (2014)

[48] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. "Can homomorphic encryption be practical? " *Proceedings of the 3rd ACM workshop on Cloud computing security workshop* 21 Oct. 2011: 113-124.

[49] https://www.schneier.com/blog/archives/2009/07/homomorphic_enc.html

[50] http://www-03.ibm.com/press/us/en/pressrelease/42808.wss

[51] There are services that aim to assist with this, e.g. https://keybase.io/

encryption is also difficult, for instance, though encrypted email systems have been around for many years, they are not yet mainstream.[52]

Trusted third-parties can play an important role in Internet communications. For instance, *Certificate Authorities* are trusted to verify—by producing *digital certificates*—that a public key belongs to the named principal. Such certificates are used, for example, in *Transport Layer Security (TLS)* to establish encrypted communication channels to protect against eavesdropping, e.g. when web browsers exchange data with web servers using HTTPS.

It is important to note that the level of security that an encryption mechanism offers is determined by both the strength of its cryptographic algorithm and its practical implementation.[53] Key length is often flagged as the main measure of strength, but this is primarily against *brute force attack,* which entails discovering a key by systematically attempting various key combinations.[54] Further, there are occasions when a broken cryptographic mechanism or implementation (such as the Heartbleed vulnerability in the OpenSSL library[55]) is discovered. This is of real concern, because a failure, or far more commonly, a compromised key, at *any* time in the future can result in potentially a wealth of data becoming vulnerable.[56] For this reason we have long argued that even the distribution of copies of encrypted data should be restricted as much as possible, to protect against possible compromises in the future.[57]

7. Managing information flows

Those using cloud services place a great deal of trust in their provider, as they must rely on the provider to manage their data properly. As such, there is ongoing research into mechanisms for managing information flows, which is driven by the need for cloud users to have some means for insight, and preferably control, over where their data flows once in the cloud.

Such work complements the control mechanisms described. Specifically:

1. In current systems, after the above-mentioned security checks (authentication and authorisation) have been carried out, there is no further control of where data can flow in a system.
2. Encryption is a mechanism that enables control over data when outside of one's physical control; though key management can be difficult, computational operations generally require unencrypted data, and distribution of encrypted data should still be managed.

The concept involves coupling data with its associated management policy, such that the management policy is known and can be enforced wherever the data flows. This is implemented by *tagging* data, where tags – representing particular management aspects/concerns – are effectively 'stuck' to data. The policy is enforced when it reaches particular parts of the system, for instance, between system processes, or as data moves from one machine to another. Enforcement of the policy may involve allowing or preventing a particular information flow, or potentially transforming data. All of this: the semantics of the tags, tag management, where enforcement occurs (both in terms of the system-stack and the position within a distributed system), and the possible policy actions, vary depending on the particular model. Importantly, this approach means that data management policy need not be encoded within the application-logic of software, but rather enables a policy separation that can apply across different infrastructure.

There are two (somewhat overlapping) categories of research in the area: taint tracking and information flow control (IFC).

---

[52] Google and Yahoo! have recently announced end-to-end encryption capabilities for their webmail services: http://www.theguardian.com/technology/2014/aug/08/google-search-results-secure-internet

[53] There are side-channel attacks in which information can be inferred without decrypting the information itself, for some examples see: http://www.cryptography.com/technology/dpa/dpa-research.html

[54] We have seen recommended encryption standards change over time, e.g. 56-bit DES keys given advancements in computational power: "Hackers prove 56-bit DES is not enough". *InfoWorld*: 77. 30 June 1997.

[55] http://heartbleed.com/

[56] There are mechanisms to mitigate loss in such circumstances, but these are not without overhead.

[57] N 42.

7.1 Taint tracking

Taint tracking involves tagging data as it flows throughout a system, where data is 'tainted' (tagged) in accordance with the properties of the system-elements it touches. Policy is enforced, based on taint, at defined points in the system, and might operate to log some information, prevent a flow, or transform data. For example, a socket receiving user input might taint the data as 'unsanitised', to stop it from entering a database until it has gone through a validation process. There is no control over the flow of data, except in accordance with policy at the enforcement points.[58]

The CSN middleware[59] provides a taint tracking approach for the cloud where tenants cooperate with each other in order to detect data leakages in the cloud platform that may affect all of them. Each tenant application includes security tags on a subset of its normal client requests, which are then logged by a set of software network monitors, deployed throughout the cloud environment by the tenants. When data that is tagged by one tenant application breaches the isolation boundary, in that the data of this tenant is observed by another tenant's monitor, it indicates data leakage.

CloudFence[60] is an approach which uses fine-grained data flow tracking to enable cloud users to independently audit their data, and allows tenants to restrict the flow of sensitive data to certain areas of the system, e.g. to limit the propagation of market data to some well-defined database or file. Silverline[61] provides *data isolation* and *network isolation* through data flow tracking within the operating system, and an enforcement layer within the hypervisor (Xen or VMWare). Tenants can taint data, such that the enforcement mechanism ensures that the data will not flow to VMs of other tenants or an untrusted location outside of the cloud provider infrastructure. This prevents leakage even where infrastructure is misconfigured.

7.2 Information Flow Control (IFC)

Information Flow Control (IFC) [62] is a data flow control model where policy is enforced against *every flow in the system.*

Specifically, IFC involves attaching labels—sets of tags representing control policy—to data, which remain with the data as they flow, in order to track and to potentially limit their propagation. Labels are also associated with principals,[63] such that enforcing protection policy involves comparing the label(s) of the data with the security context of the principals. A flow is allowed only if the label on the data and security context of the principal accord. Traditionally, IFC entailed a universal, fairly-static label set; *Decentralised Information Flow Control (DIFC)*[64] builds upon this, bringing more flexibility as it allows the dynamic introduction of new labels. This allows systems to evolve in line with new classes of security concern, keeping existing policy intact.

Policy is enforced against every flow in the system, which is made possible by the fact that each principal runs in a security context, defined by its labels. The side effect is that this also facilitates a more complete audit, as the policy decision for every flow can be logged, as well as the security contexts for the principals involved.

---

[58] One issue that taint-tracking systems must manage is that data can continually accumulate taint such that it becomes generally unsable.

[59] Christian Priebe et al. "CloudSafetyNet: Detecting Data Leakage between Cloud Tenants" *Proceedings of the 2014 ACM workshop on Cloud computing security workshop* Nov 2014: *to appear*.

[60] Vasilis Pappas et al. "CloudFence: Data Flow Tracking as a Cloud Service." *Research in Attacks, Intrusions, and Defenses* (2013): 411-431.

[61] Yogesh Mundada, Anirudh Ramachandran, and Nick Feamster. "Silverline: Data and network isolation for cloud services." *Proc. of HotCloud* (2011).

[62] Dorothy E. Denning "A lattice model of secure information flow." *Communications of the ACM* 19.5 (1976): 236-243.

[63] N 40.

[64] Andrew C Myers, and Barbara Liskov. *A decentralized model for information flow control*. ACM, 1997.

At Cambridge, we have been developing infrastructure for IFC-enabled cloud services. We have designed an importable kernel[65] module (FlowK)[66] that intercepts system calls[67] and enforces IFC by label checking. This enables the transparent enforcement of IFC policy at the operating system level, on data flows between system processes. We have also implemented an IFC enforcement mechanism within a messaging middleware (SBUS). The motivation is to enable the granular control of flows between systems, at a higher level of abstraction, i.e. at the level of named, typed attributes for individual messages (e.g. someone's date-of-birth could be tagged to have different security properties (labels) from their surname). Integrating FlowK and SBUS represents a step towards achieving end-to-end IFC enforcement, as it enables enforcement both within and across machines.

Note that we are focused on the mechanism for enforcement; policy is defined at higher levels. Our architecture provides application managers, software tasked with the function of setting up security contexts (labels and privileges) for other software (applications and services).

7.3 Potential for the cloud

We feel that such research, particularly IFC, offers promise in terms of compliance and control.

Firstly it provides an extra level of protection for shared infrastructure, improving service provision and increasing levels of trust. Though IaaS already isolates tenants (when properly implemented), there will be situations that require collaboration across IaaS services (i.e. tenants sharing data). IFC provides the means for managing and securing information flows both within and between virtual machines. For some PaaS offerings, there is strict isolation between tenants (e.g. Linux containers), but this is a complete separation and thus entails resource duplication and can hinder data sharing. IFC could provide a suitable security abstraction, where mandatory security checks occur at the interfaces between the software components of the PaaS platform, such as interactions with databases, messaging systems, etc., and including interactions between tenants. Providers offering IFC-enabled SaaS applications could increase user confidence that their data is being compartmentalised correctly, and give some insight into how their data are used and managed by the service.

Data flow-focused policy enforcement results in detailed logs. These have the potential to *demonstrate compliance with laws or with contracts* between cloud providers and users. Strong trustworthy audit logs might also work to help instil trust in the enforcement mechanism, and also help identify and resolve policy errors, e.g. where a legitimate flow (in terms of control policy) results in a situation not previously considered.

IFC allows control policy to be separated from implementation specifics. Thus labels, apart from containment, can be used to realise a variety of higher-level policy concerns.[68] One could imagine SaaS applications allowing end-users to add custom constraints to their data, automatically or through some interface, e.g. that certain data is personal and thus must be subject to greater protections. Labels could also be used to enable the concept of *"virtual jurisdiction"*[69], by reflecting the laws by which the data should be governed. In this way, data could flow only to persons (their applications) and providers assured to be subject to the required laws.[70] It would also allow several jurisdictions to exist in the same physical machine, with the transfer of information between jurisdictions being an explicit action through a trusted mechanism that is securely logged for future audit. IFC allows for nuanced control as particular flows are regulated as appropriate, as opposed to complete isolation.

---

[65] A *kernel* is a part of the OS that mediates between software (processes) and the computer's resources. The kernel is accessed through system calls (n 67), which relate to software I/O requests.

[66] Thomas F. J.-M Pasquier et al. "FlowK: Information Flow Control for the Cloud" *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference* Dec. 2013: *to appear.*

[67] Software (processes) uses *system calls* to interact with an OS. This is typically to request some resources or access some services, e.g. to read from disk, communicate via a socket, start a new software process, etc.

[68] Particularly DIFC that allows the dynamic definition application or even data-subject-specific management policies

[69] See n 1.

[70] Please see the appendix in n 1 for discussion of the complexities concerning the use of the word 'jurisdiction'.

Considerations

It must be noted that though taint-tracking and IFC models show promise, their application to cloud services is still an area of active academic research. Policy enforcement entails overhead, though initial results from some of our IFC experiments indicate an average of 10% overhead in I/O heavy applications. A more general concern is that work is needed to make the use of IFC natural and easy, and though our research aims for applications to be able to use IFC without being changed, there are still issues of policy expression and audit interpretation. This includes having a scalable label-naming scheme (perhaps DNS-like), and mechanisms for automatic translation of application policy into labelled entities.

An important consideration, for any policy-enforcing technology, is that the enforcement mechanism must be *trusted* to enforce the policy. Towards this end, an explicit design goal of IFC work is to make the trusted aspect as small as possible, and perhaps shared across various operational and administrative domains. The degree of trust depends on the level at which the enforcement mechanism applies. For instance, FlowK as a kernel module ensures protection at the OS-level above, but as such, assumes a trustworthy hypervisor and hardware. That said, there is recent work in leveraging new hardware-based isolation techniques that aim directly at these issues of trust in a cloud-context.[71] Thus one can envisage third-party attestation services, and hardware-based evaluation infrastructure, that work to demonstrably ensure the enforcement of data flow policy.

8. Conclusion

We have provided an overview of some of the technical mechanisms enabling cloud governance, and how these relate to the concerns surrounding regional clouds. To summarise, in the context of a hypothetical Europe-only cloud, routing could be configured (e.g. via BGP) to ensure that traffic remains within Europe, assuming the relevant infrastructure providers agree and/or are compelled to do so. Such an approach is rather blunt; being service-level it leaves users with no means for control, and applies to all traffic, irrespective of the data (and their associated constraints). Advances in cloud provisioning enable management at higher-levels: improvements in the trusted computing base aids assurance, and unikernels allow targeted, tracked and easily deployed VMs that could be constructed to encapsulate regional and/or jurisdictional concerns. Often different data will have different management constraints. Concerning data-centric governance, access controls protecting data should operate beyond the point of interaction between users and the service. Encryption is useful, though key management is difficult, and in the Europe-only case, jurisdiction over the certificate authorities, key-escrow services, attestation services, etc., would be important. We feel that IFC shows promise for cloud computing as it enables data flows to be controlled and audited across system-level boundaries in accordance with defined constraints, which could include geographic or jurisdictional aspects, amongst others.

It is clear that issues of accountability, transparency and control lie at the heart of concerns regarding the cloud, and the Internet more generally. There is no technical silver-bullet; policy-making processes, international negotiations, consumer power, and so forth, will be crucial in resolving these. The technical mechanisms described in this paper address particular aspects of transparency and control, at different technical-levels and in different ways. However, the implementation of such mechanisms is not without cost. If data management policy, reflecting user, legal, economic and other higher-level concerns, can be agreed and defined, these technical tools can be combined to improve governance, and thus increase trust in cloud services.

This should encourage the uptake and use of cloud services—extracting more value and benefit from the technology, and encouraging further investment in its development. At the same time, the technical protection mechanisms will work to influence and shape the development of such policy, and indeed, the higher-level considerations that the policy encapsulates.

[71] See K. R. Jayaram, et al. "Trustworthy Geographically Fenced Hybrid Clouds" *In ACM/IFIP/USENIX 15th International Middleware Conference*. Dec. 2014: *to appear,* and n 31 for examples.