Know your cirrus from your cumulus

how clouds differ beyond cost and speed and how that affects you

jclouds.org
github.com/jclouds/jclouds

jclouds

jclouds.org
github.com/jclouds/jclouds

jclouds

These clouds…

jclouds.org
github.com/jclouds/jclouds

jclouds

…all look…

different.

Does it matter?

Well, it does to me! Let me give you an example. We were flying towards what we thought…

…was a nice “puffy”, but when we hit it at 6000ft or so…

jclouds.org
github.com/jclouds/jclouds

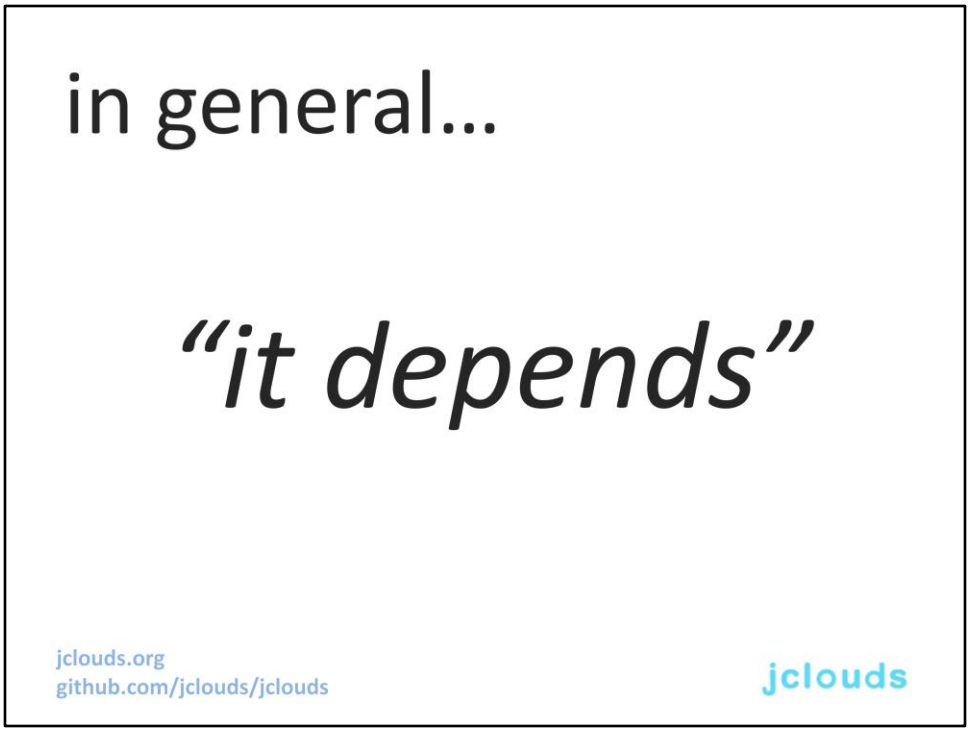
jclouds

…it turned into something like this. 3000ft of solid cloud bank.

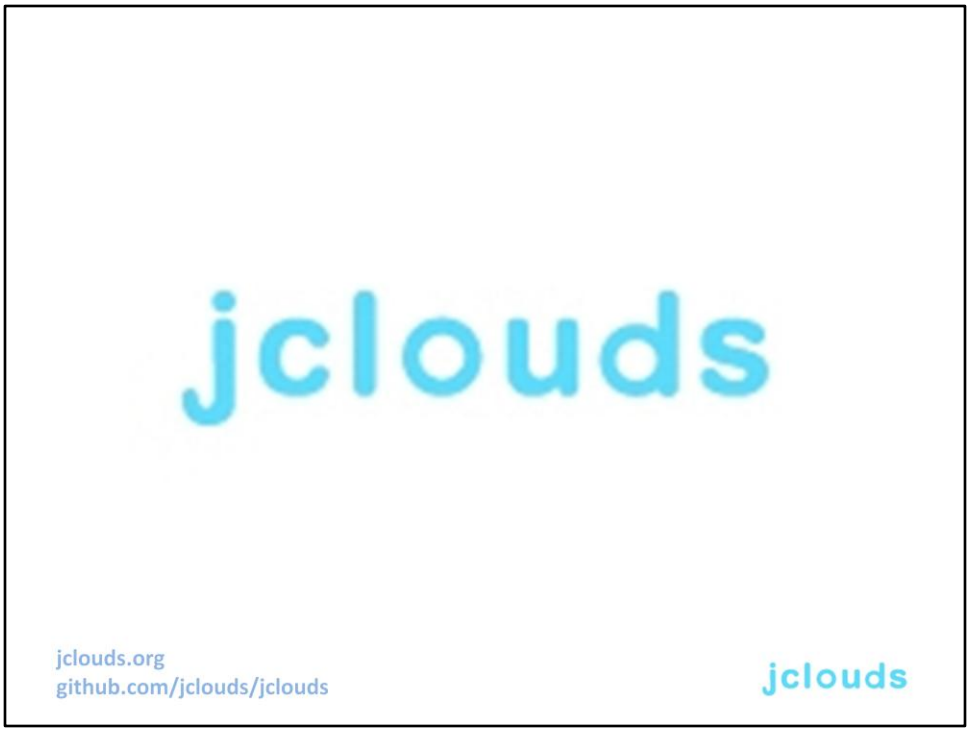
When we finally got below it, we were here.

For the record, we made it quite a long way back, and everyone was safe.

In general, whether the difference between clouds should matter to you is an open and very good question.



The Architects’ Answer applies: “it depends”. Opinions differ widely on this – as I was preparing this talk, we had some very interesting ideas about that.



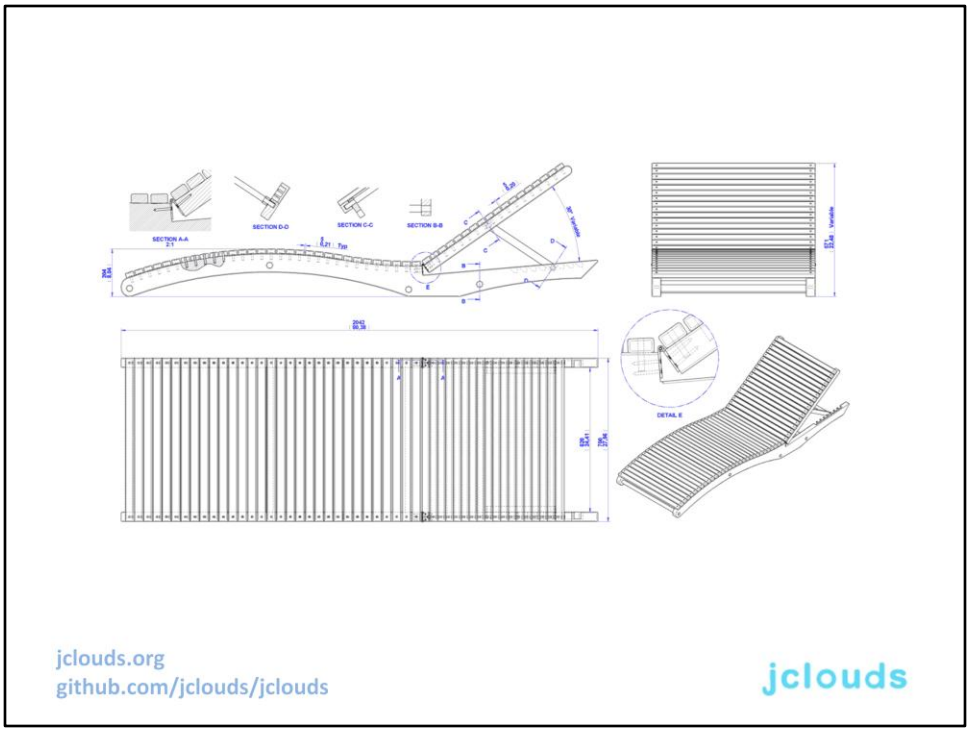
But how did this whole thing come about in the first place? Well, I spend some of my night hours on jclouds, the Java multi-cloud library.

jclouds connects to a whole bunch of different cloud providers all the way from IaaS to SaaS, and we have users from every part of the cloud spectrum. From…

jclouds.org
github.com/jclouds/jclouds

jclouds

…people that *build* platforms…



…to people that build *service offerings* on top of platforms…

jclouds.org
github.com/jclouds/jclouds

jclouds

…to *users* of those service offerings.

So we get lots of feature requests, some pretty much ahead of the curve with respect to the rest of the cloud landscape.

jclouds.org
github.com/jclouds/jclouds

jclouds

And as we were getting some many requests for additional metadata beyond what the main jclouds Compute and BlobStore APIs provide, we – Adrian, mainly – embarked on an intensive round of information and opinion gathering to see how to best support our users.
As expected, views differed quite widely, even amongst experts. From…

jclouds.org
github.com/jclouds/jclouds

jclouds

…”why should you care” to…

jclouds.org
github.com/jclouds/jclouds

jclouds

…”I need full access”.

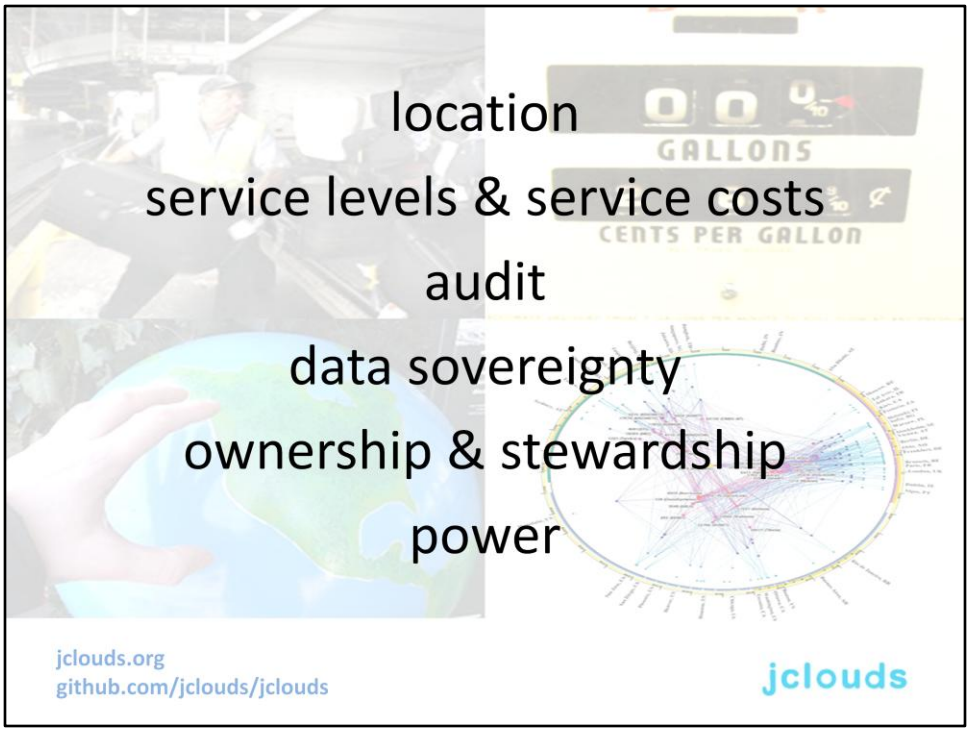
So I should make it very clear now that I am **not** about to present the One Jclouds Truth about cloud metadata and how much you should have access to or need to have access to.
Rather, this is a summary and analysis of what we discovered there, and what we've done with this information so far.

This is not about you accepting some "gospel truth"…

…but about putting you in a position to

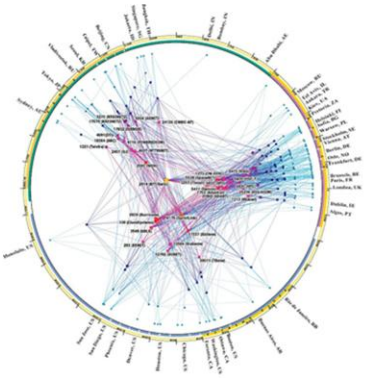
a) know about some of the non-obvious issues out there
b) have an idea about how these issues may or may not impact your applications and business
c) know where to go to get the information you need

In other words, at the end of this talk, you should be further along the road to being able to make an informed decision about how much you want/need to care about cloud details, and how to architect for that!



The points we'll address today…

Unsurprisingly, given that the topics came from our users' use cases, many are related to the "classic" cloud topics around performance, traceability, jurisdiction, security and recovery.



jclouds.org
github.com/jclouds/jclouds

jclouds

Let's start with location. First of all, it's important to distinguish between *physical* (i.e. *geographical*) and *logical* (i.e. *network*) location.

jclouds.org
github.com/jclouds/jclouds

jclouds

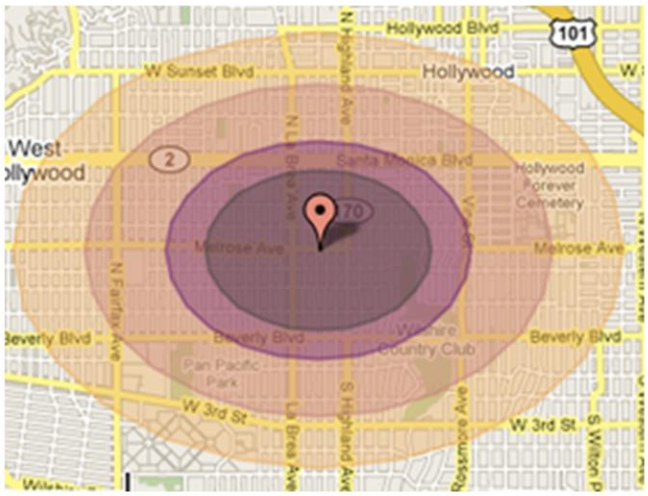
If you're talking *physical* location, sometimes it's the *absolute* location you're talking about.

Say, within a certain number of hours drive from a certain point, or within a certain geographical area (think jurisdiction).

jclouds.org
github.com/jclouds/jclouds

jclouds

Sometimes, it's only *relative* location that you're interested in.



jclouds.org
github.com/jclouds/jclouds

jclouds

Such as – and yes, using the *same* slide to illustrate ;-) – not within 100 miles of each other (think blast radius and disaster recovery) or within 1h drive of each other (think engineer on an emergency callout).
Challenges:
- what kind of representation to choose? Lat/long not offered by enough providers, plus may vary over time, plus impossible to define for the SNIA concept of “set of regions” where there is no one point.
- Some providers have a “location” model that doesn’t correspond to jurisdictional boundaries or “real-world” political entities, which is important for almost all “physical location” use cases.
- Some kind of manual mapping required, but luckily the physical location of the actual datacenter(s) – if known – is pretty damn static info.
- Good compromise: ISO-3166

Other times, you're not really interested in the *physical* proximity at all, but in the *network* proximity, which is mainly interesting in terms of *performance*. Such as: connected by a network connection of at least a certain speed, or within the same "performance or SLA domain" (e.g. availability zone). Or ping times, bandwidth guarantees, speed to main internet peers etc.
Or even *relative logical location*, i.e. "behind a firewall" or "in the same datacenter".

Challenges:
- How to choose "representative" IPs when these might shift?
- How to measure speed/performance of these IPs in a consistent way when some providers e.g. de-prioritize ICMP traffic

jclouds.org
github.com/jclouds/jclouds

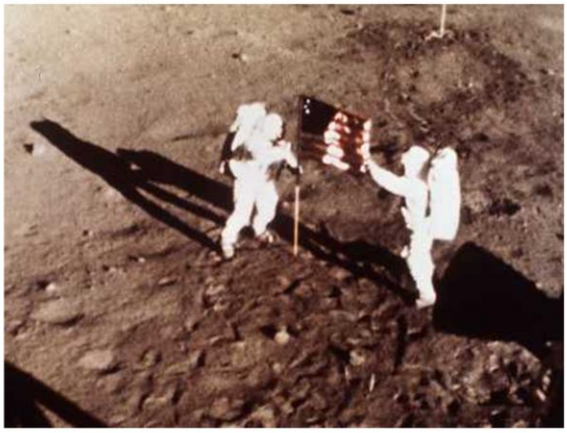
jclouds

As we see, many location use cases boil down to jurisdiction, which has relationships also to audit, ownership, stewardship, data sovereignty. This is a hard problem – there are already some startups attempting to help you through this.

jclouds.org
github.com/jclouds/jclouds

jclouds

Many companies require audit, but there are as yet few standards or means to identify audit levels of providers, let alone verify them. Cloudaudit is trying to tackle this.
Note that audit is not the same as security, although both are “cross-cutting concerns” in the sense that they affect all your choices of platform and service providers, but require different hooks. And, depending on the actual technology you’re talking about, these hooks may not even exist…but more about the problem of “layers” in a bit.



jclouds.org
github.com/jclouds/jclouds

jclouds

Ownership of data remains important, but it is not enough. Of course, in most cases your data is still *yours*, but suddenly you are dependent on…

jclouds.org
github.com/jclouds/jclouds

jclouds

…a steward to handle this data for you responsible. It's no longer your house, and no longer your "house rules", either. I.e. this steward may well have policies that…

jclouds.org
github.com/jclouds/jclouds

jclouds

…prevent him from doing so! And, depending on your offering, there may be multiple "nested" stewards involved – think CloudBees using EC2, all of whom have policies you may be violating. And, in fact, in many cases you may not even be *aware* of these nested stewards because your immediate provider may not expose that implementation detail, or may decide to switch.
The WikiLeaks story is a good demonstration of this…

≠

jclouds.org
github.com/jclouds/jclouds

jclouds

This can mess with you especially if you try to apply a “traditional” steward model such as UPS. Of course, they *also* have T&Cs that prevent you from shipping certain items, but those items remain *yours* and, more importantly, they *remain*. In the cloud, if your provider takes you down, your data may be *gone*.

In general, there are many questions around data sovereignty and relations to handling policy, especially how to enforce this policy programmatically?
- can I control where my data goes?
- can I set bounds on what is happening where?
- clients with zero tolerance on data location (e.g. us-standard)

Don't be fooled – big companies are actively looking at this, even if they may not talk about it.

jclouds.org
github.com/jclouds/jclouds

jclouds

SLA and pricing information is naturally a key metadata concern, especially for platform providers that wish to implement automated chargebacks and/or make intelligent, automated choices about load balancing and system provisioning. Key challenges here:
- defining a standard “compute unit” and pricing model for that unit
- defining a service level descriptor that can be compared
-define a comparable “rebate policy”

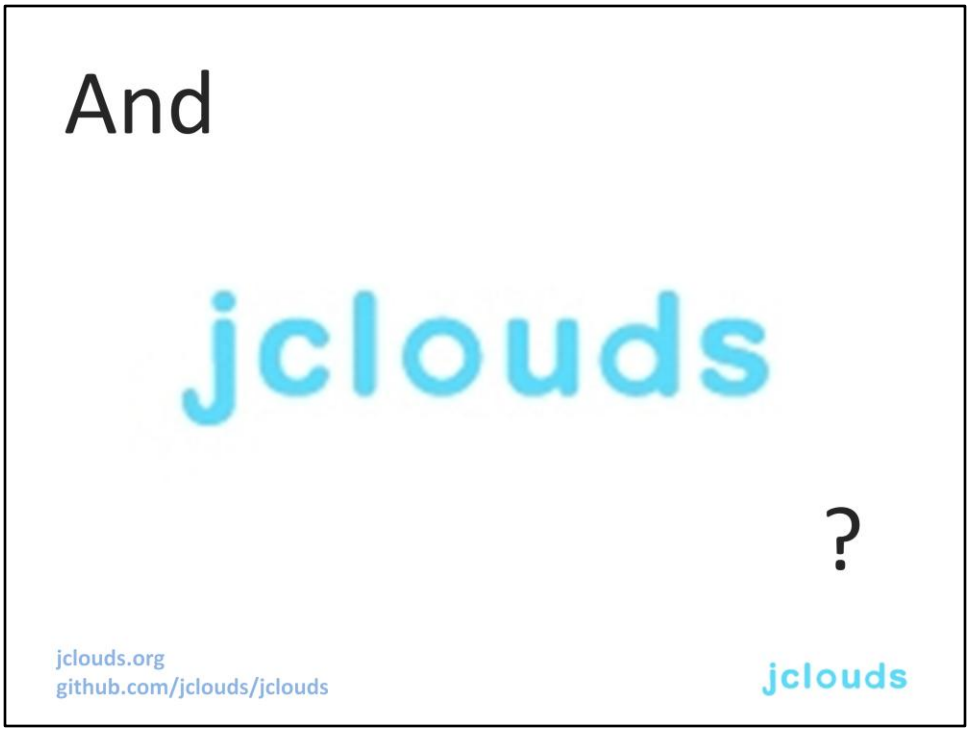
Note that service levels in the cloud can be more tricky than at first might seem – being inaccessible (perhaps even just being inaccessible from a certain *location*) may not qualify as a violation of service level. See the EBS case.

In other words, if you’re concerned about this it’s wise to take a chaos monkey approach and assume everything can become unavailable.

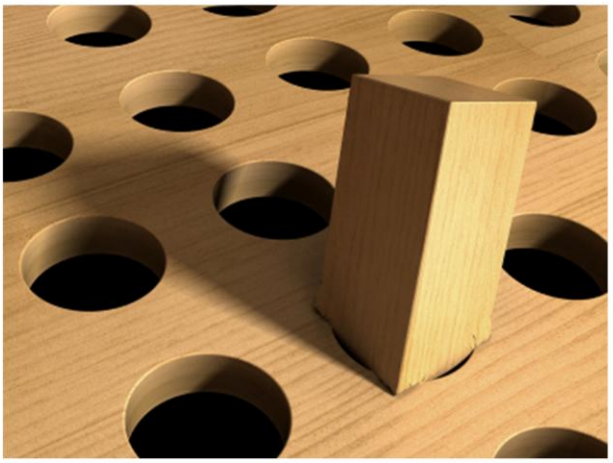
jclouds.org
github.com/jclouds/jclouds

jclouds

Power is still something of a differentiator at present (e.g. “green” vCloud provider), but may become a legal requirement in some areas, e.g. Cali. Note that providers appear unlikely to want to expose this kind of potentially sensitive data to external services.



So what did jclouds take away from these inputs?

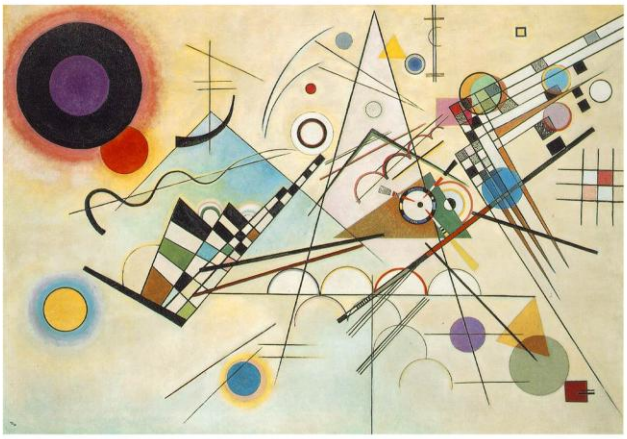


jclouds.org
github.com/jclouds/jclouds

jclouds

Well, it's a hard problem! There are plenty of complicating factors:
- missing or incompatible hooks

- missing data



jclouds.org
github.com/jclouds/jclouds

jclouds

- data that is not automatically parsable

jclouds.org
github.com/jclouds/jclouds

jclouds

A fundamental issue across *all* these metadata domains relates to the more-and-more-frequently violated assumptions around the layering of cloud infrastructure (host OS, hypervisor, guest OS etc.).

jclouds.org
github.com/jclouds/jclouds

jclouds

We've tried to come up with nice architectural abstractions here, that layers represent some kind of infrastructure or software platform (hypervisor, guest OS etc.)…

jclouds.org
github.com/jclouds/jclouds

jclouds

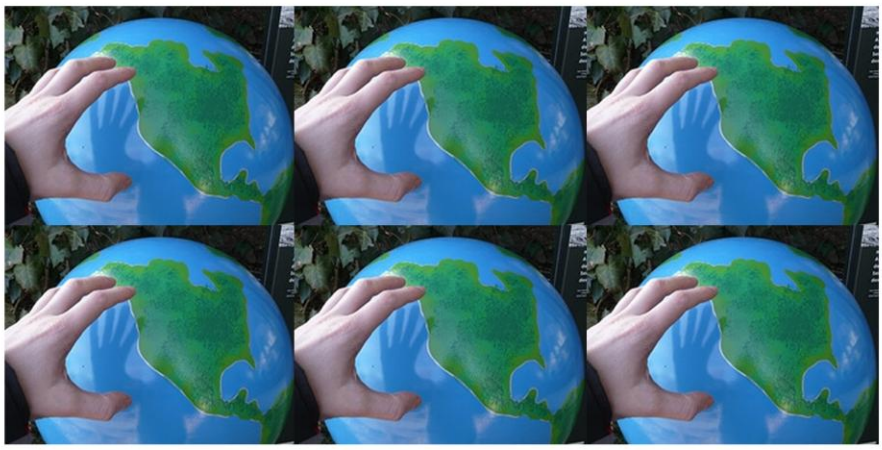
…but in many cases this is a bit of a magic trick…

jclouds.org
github.com/jclouds/jclouds

jclouds

…and sometimes it goes wrong ;-)

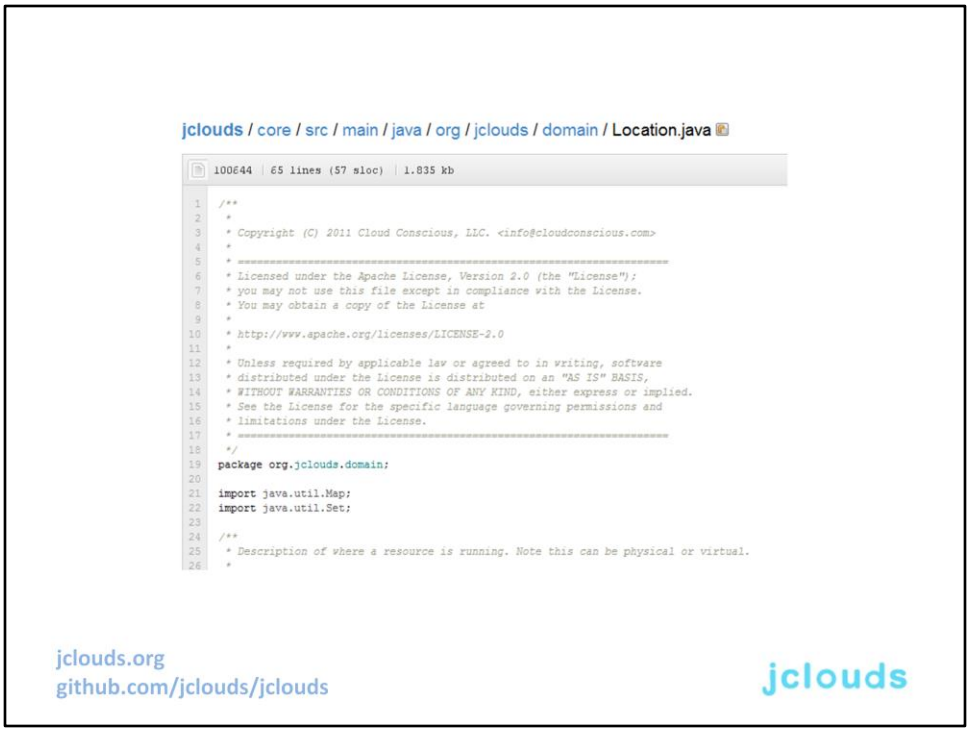
There are plenty of offerings out there that nest these (e.g. CohesiveFT and CloudSwitch with nested hypervisors) or virtualize/simulate others, meaning that hooks that expect to be able to access the “layer below” may not work – or worse, may transiently produce different results.



jclouds.org
github.com/jclouds/jclouds

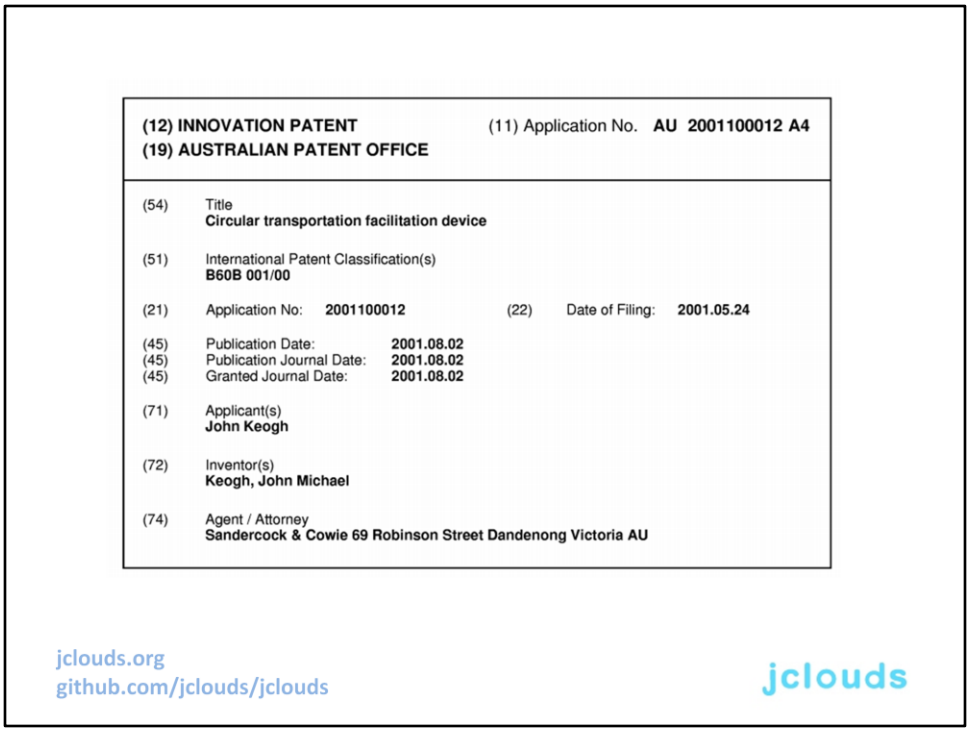
jclouds

Conclusion:
Small wrinkles (e.g. us-standard, us-west but no us-east) aside, location is both fairly well definable and appears in *many* use cases



So jclouds introduced the Location API

Others still too vague/too non-standard/too much “null” data/too much manual work/not enough use cases



(12) INNOVATION PATENT (11) Application No. **AU 2001100012 A4**
(19) AUSTRALIAN PATENT OFFICE

(54) Title
Circular transportation facilitation device

(51) International Patent Classification(s)
B60B 001/00

(21) Application No: **2001100012** (22) Date of Filing: **2001.05.24**

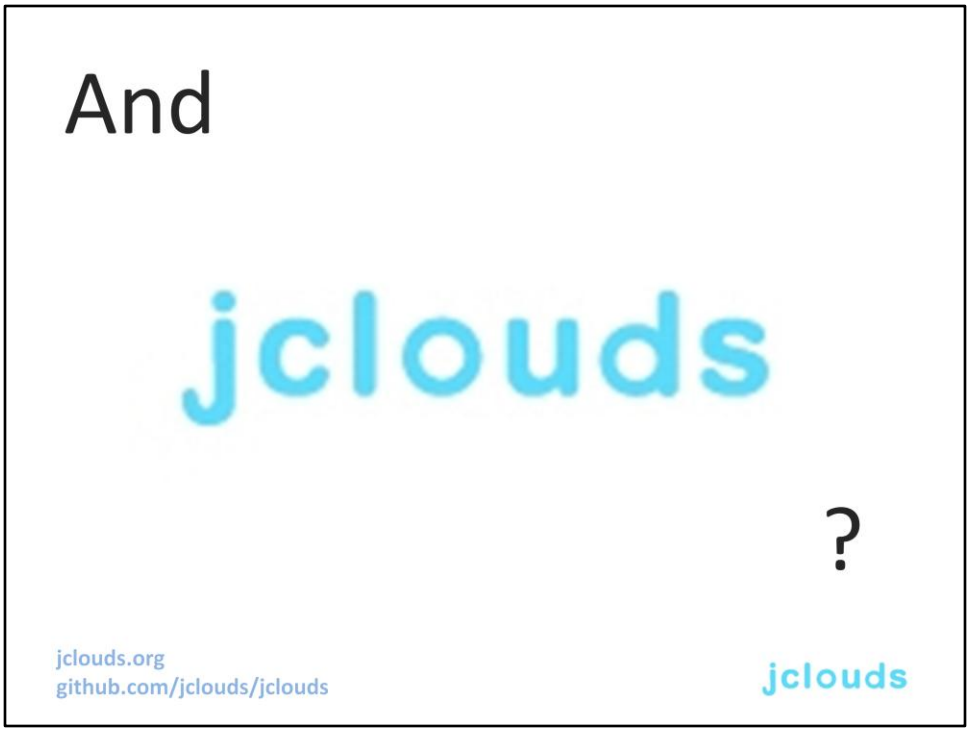
(45) Publication Date: **2001.08.02**
(45) Publication Journal Date: **2001.08.02**
(45) Granted Journal Date: **2001.08.02**

(71) Applicant(s)
John Keogh

(72) Inventor(s)
Keogh, John Michael

(74) Agent / Attorney
Sandercock & Cowie 69 Robinson Street Dandenong Victoria AU

We don’t want to reinvent any wheels here – if possible, one can look at delegating to others e.g. cloudaudit.



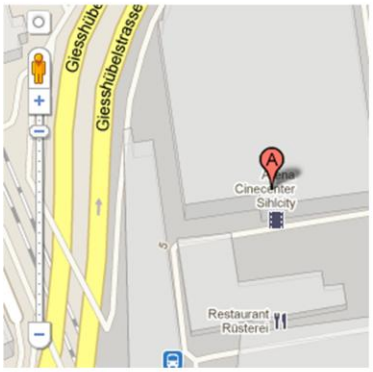
Of course, other organisations are looking at these metadata issues too – e.g. SNIA's model of allowing the *client* to express constraints on the provider they want, rather than simply exposing the data and getting the client to choose ('Data System Metadata')

Open question: not too complicated for storage providers to offer?

The OSGi Cloud Working Group is also looking at these issues, and e.g. deltacloud also supports capability metadata, although from a client-query perspective

jclouds.org
github.com/jclouds/jclouds

jclouds



jclouds.org
github.com/jclouds/jclouds

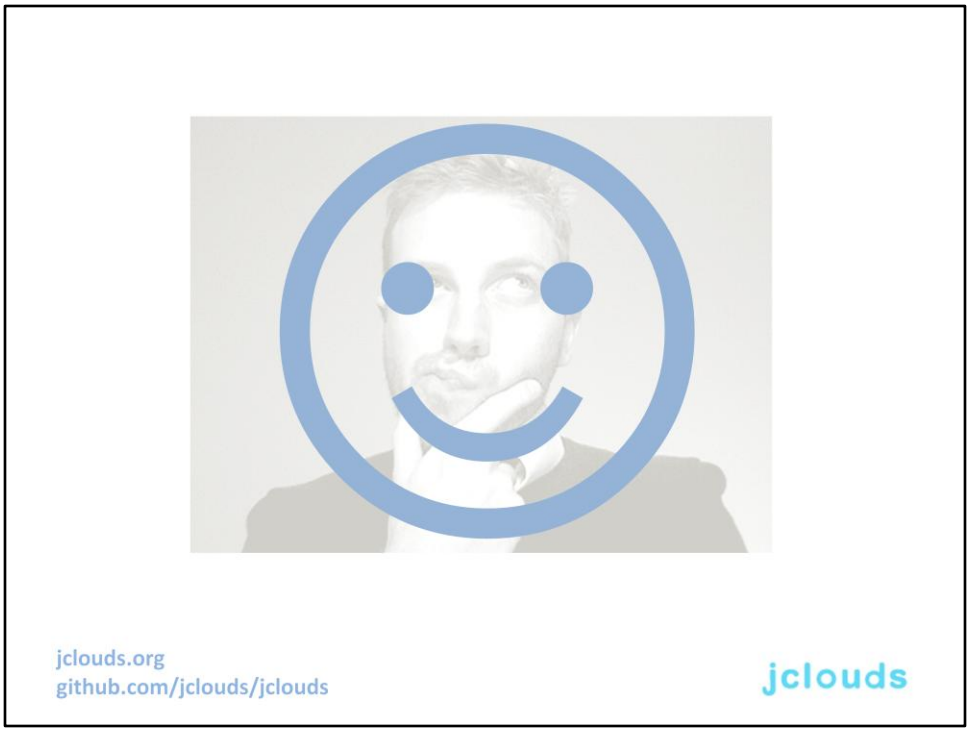
jclouds

Basically, whether you need to care about a lot of this depends on the “zoom level” you want to choose. Of course, this decision may change over time – witness lots of people now thinking about having fallback options if EC2 goes down again…

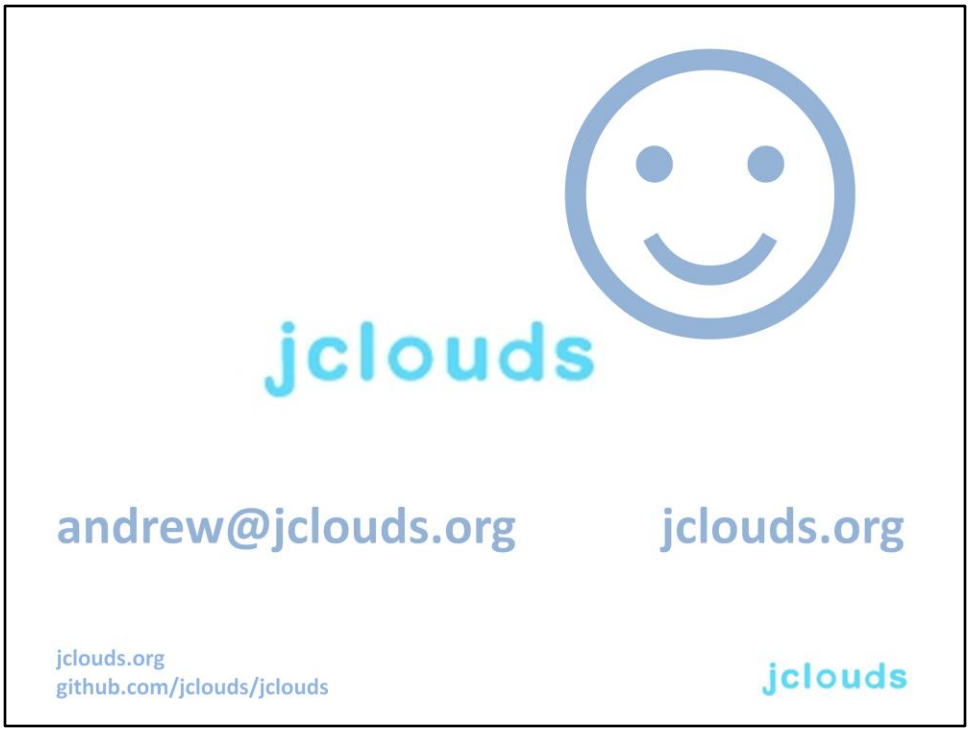
jclouds.org
github.com/jclouds/jclouds

jclouds

…but hopefully you're now at least in a position to *think* about these issues…



…in a useful and constructive way.


jclouds
andrew@jclouds.org
jclouds.org
jclouds.org
github.com/jclouds/jclouds
jclouds

Thanks!

http://www.slideshare.net/apwashere/know-your-cirrus-from-your-cumulus
(as soon as we're done)

http://www.slideshare.net/jclouds/location-matters-even-in-the-cloud-cloudcamp-london
http://code.google.com/p/jclouds/wiki/LocationMetadataDesign
http://code.google.com/p/jclouds/issues/detail?id=418

http://cdmi.sniacloud.com/
(esp. 16.4 *Data System Metadata* and 16.6 *Provided Data System Metadata*)
https://www.osgi.org/bugzilla/show_bug.cgi?id=114

jclouds.org
github.com/jclouds/jclouds

jclouds