Report from Dagstuhl Seminar 15072

Distributed Cloud Computing

Edited by

Yvonne Coady[1], James Kempf[2], Rick McGeer[3], and Stefan Schmid[4]

1 **University of Victoria, CA,** ycoady@cs.ubc.ca
2 **Ericsson – San Jose, US,** james.kempf@ericsson.com
3 **HP Labs – Palo Alto, US,** rick@mcgeer.com
4 **TU Berlin, DE,** stefan.schmid@tu-berlin.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 15072 "Distributed Cloud Computing". A distributed cloud connecting multiple, geographically distributed and smaller datacenters, can be an attractive alternative to today's massive, centralized datacenters. A distributed cloud can reduce communication overheads, costs, and latency's by offering nearby computation and storage resources. Better data locality can also improve privacy. In this seminar, we revisit the vision of distributed cloud computing, and identify different use cases as well as research challenges.

Seminar February 8–11, 2015 – http://www.dagstuhl.de/15072
1998 ACM Subject Classification C.2.4 [Computer-Communication Networks] Distributed Systems
Keywords and phrases Distributed Cloud Computing, Distributed Systems
Digital Object Identifier 10.4230/DagRep.5.2.64
Edited in cooperation with Oliver Hohlfeld

1 Executive Summary

Yvonne Coady
James Kempf
Rick McGeer
Stefan Schmid

The Dagstuhl Seminar on Distributed Cloud Computing was held Feb. 8–11, 2015. 22 researchers attended the multidisciplinary seminar from the areas of networking, cloud computing, distributed systems, operations research, security, and system administration. In contrast with the centralized cloud deployment model where applications are restricted to a single mega-data center at some network distance from the customers, in the distributed cloud deployment model, many smaller data centers are deployed closer to customers to supplement or augment the larger mega-data centers, and the smaller data centers are managed as one pooled resource. Two administrative models of a distributed cloud are common today: the integrated model where a single administrative entity controls all the data centers and the federated model where multiple administrative entities control the data centers and users authenticate for resource access using a federated identity management system. Over the course of the 3 day seminar, 15 presentations were given on various aspects of distributed cloud or the disciplinary areas relevant to distributed cloud. The

Distributed Cloud Computing, *Dagstuhl Reports*, Vol. 5, Issue 2, pp. 64–79
Editors: Yvonne Coady, James Kempf, Rick McGeer, and Stefan Schmid
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

seminar shared two talks with the concurrent seminar on Foundations of Networking and attended one of the Foundations of Networking Talks. Taking the presentations as input, the seminar then broke into three groups to discuss a research agenda for distributed cloud. The groups were requested to come up with 3 questions in their particular area (distributed systems, programming models, and cloud) and two for the other two groups. At the end of the seminar, the group discussed forming a research community around distributed cloud with an annual conference. Currently, a workshop on distributed cloud is held annually, called DCC (for Distributed Cloud Computing). This year's workshop will be held in conjunction with SIGMETRICS in Portland, Oregon in June. Slides, abstracts of the talks and reports from the breakout groups are available in the Dagstuhl content management web site. An extended version of this report appeared in the April 2015 issue of ACM SIGCOMM Computer Communication Review [1].

References

1 Yvonne Coady, Oliver Hohlfeld, James Kempf, Rick McGeer, and Stefan Schmid. Distributed cloud computing: Applications, status quo, and challenges. *SIGCOMM Comput. Commun. Rev.*, 45(2):38–43, April 2015. DOI: 10.1145/2766330.2766337

2 Table of Contents

Working Groups Identifying Research Challenges

3 Overview of Talks

3.1 Scalable consistency – all the way to the edge!

Annette Bieniusa (TU Kaiserslautern, DE)

Joint work of Zawirski, Marek; Bieniusa, Annette; Preguica, Nuno; Duarte, Sergio; Balegas, Valter; Shapiro, Marc
Main reference M. Zawirski, A. Bieniusa, V. Balegas, S. Duarte, C. Baquero, M. Shapiro, N. Pregiuça, "SwiftCloud: Fault-Tolerant Geo-Replication Integrated all the Way to the Client Machine," Research Report RR-8347, HAL ID hal-00870225, 2013.
URL https://hal.inria.fr/hal-00870225

Distributed cloud computing allows to move the execution of distributed applications towards client machines. Current data management solutions for cloud infrastructures replicate data among several geographically distributed data centres but lack support for managing data maintained by clients. This talk presents SwiftCloud, a storage infrastructure for cloud environments that covers this gap. SwiftCloud addresses two main issues: maintaining replicas consistent and maintaining client replicas up-to-date. SwiftCloud pushes the scalability and concurrency envelope, ensuring transactional causal consistency using Conflict-Free Replicated Data Types (CRDTs). CRDTs provide higher-level object semantics, such as sets, maps, graphs and sequences, support unsynchronised concurrent updates, while provably ensuring consistency, and eschewing rollbacks. Client-side replicas are kept up to date by notifications, allowing client transactions to execute locally, both for queries and for updates.

3.2 Seattle Testbed (overview)

Justin Cappos (New York University, US)

URL https://seattle.poly.edu

Traditional distributed computational models, such as client-serverand cloud computing, involve moving computation from geographically distributed devices with little computational power to well-provisioned centralized servers. In this work, we explore the idea of harnessing computational resources on end user devices in an on-demand, cross application manner. Using this paradigm, we have constructed the Seattle testbed. Seattle makes it practical for arbitrary Internet users to securely participate in our testbed without compromising the security or performance of their laptop, desktop, phone, tablet or other device. Seattle has been deployed for six years across tens of thousands of end user devices. Seattle has wide spread practical use as a testbed for researchers and educators, including use in more than 50 classes at two dozen universities. The talk will include a demo of the system described.

3.3 A few Words on Data Management

Lars Eggert (NetApp Deutschland GmbH – Kirchheim, DE)

A brief overview of what I mean by data management, as well as some storage trends.

3.4 IoT Meets Cloud – Now we have to work out the details

Johan Eker (Lund University / Ericsson Research, SE)

In a not too distant future we expect to have tens of billions of devices connected to cloud and some of them will be providing services that require predictable latency and high availability. The cloud of tomorrow will move beyond today's IT services and into mission critical areas such as automation and health-care. To explore the full potential of such a scenario we must provide a programming platform that exposes cloud services and network functionalities in a simple and straightforward matter. This talk will present the Calvin project that aims at developing a programming framework for applications that spans over a heterogeneous hardware platform consisting of mobile sensors and cloud. This is work-in-progress and it will likely raise more questions than it answers.

3.5 The Rise of Software Defined Infrastructure

Chip Elliott (BBB Technologies – Cambridge, US)

We observe the interesting convergence between multi-tenant clouds, distributed clouds, network functions virtualization, and software defined networking, and discuss the emergence of software defined infrastructure.

3.6 Towards Federated Big Data Processing

Patrick Eugster (Purdue University – West Lafayette, US)

The "cloud" has a strong potential for efficiently performing analyses overlarge datasets. However, several hurdles need to be overcome to fulfill the vision of federated big data analyses across say patient records hosted by different parties and in different datacenters. First, copying all data to a single datacenter for subsequent analysis is inefficient, if feasible at all under sharing regulations. Second, especially when leveraging public clouds, processing confidential data in the cloud is all but desirable with the security dangers implied by multi-tenancy underlying cloud platforms. In this talk I will survey some practical first steps we made towards addressing these challenges. This includes our work on (a) geo-distributed big data analysis and (b) assured cloud-based big data analysis. In short, the former consists in moving computation towards data rather than only the other way around, and the latter consists in leveraging a combination of replication and partially homomorphic encryption to ensure integrity/correctness and privacy of big data analyzed in the cloud.

3.7 Distributed Clouds: Opportunities and Challenges for IT security

Hannes Hartenstein (KIT – Karlsruher Institut für Technologie, DE)

A security objective does typically not exist in isolation, but in combination with other objectives: confidentiality *and* performance, confidentiality *and* availability etc. In this talk/discussion we look at two use cases that both show benefits and challenges of distributed clouds, namely confidential data outsourcing and secret sharing schemes. We quickly scan through existing work that makes use of fragmentation techniques and of confidentiality preserving indexes. We show how confidentiality can be achieved based on non-colluding cloud providers and how the resulting trade-offs with performance and availability can be tuned for the cases of outsourcing databases and outsourcing strong cryptographic keys based on secret sharing schemes. The presentation should serve as a starting point for a discussion on appropriate security-related assumptions on properties of the architectures of distributed clouds.

3.8 User Perception of Distributed Clouds

Oliver Hohlfeld (RWTH Aachen University, DE)

A missing understanding of user requirements can challenge system engineering by its potential to cause over-engineered system architectures and sub optimal user experience. To foster research on improved distributed cloud architectures, this talk motivates the study of user requirements. Concrete requirements are dependent on several factors including applications, data, or different user types: humans (e.g., cloud gaming) vs. machines (e.g., virtualized network functions in carrier clouds). The talk first discusses restrictions on data storage locations as one example of potential user requirements. In this example, policy languages restrict data processing and storage in distributed clouds. It is further shown how user studies can be conducted to assess the impact of system design / network artifacts on end-user experience.

3.9 Cloud/WAN Networking

James Kempf (Ericsson – San Jose, US)

Cloud/WAN networking has become easier in recent years, but still needs work. I briefly discussed a few motivating use cases, bandwidth calendering, dynamic on the fly branch office VPN gateway deployment, telcom applications, then presented the Cloud Atlas architecture. Cloud Atlas supports orchestration of cloud/WAN network connections on any sort of wide area network virtualization technology by stitching the WAN connections into the Neutron tenant networks in an OpenStack cloud. A few simple abstractions provide the programmer with easy access to the WAN VPN. We implemented Cloud Atlas on a few different WAN VPN substrates, all supporting the MPLS L2VPN service VPLS. Dynamic on the fly branch office VPN deployment was also implemented under Cloud Atlas, with VLAN tags inside IPSec/GRE tunnels providing the WAN virtualization. Our experience with these prototypes led us to simplify the architecture down to a single primitive, the Gateway API, with bridges an L2 Neutron network to a Provider network. The Gateway API has been submitted to the Neutron working group for incorporation into the OpenStack release.

3.10 Mobile Distributed Cloud

James Kempf (Ericsson – San Jose, US)

I discussed a few representative use cases for the distributed cloud. One involves applications that have tight latency constraints (less than 50 ms). Another involves applications that have lots of hyper-local data that needs to be downloaded quickly or applications where a large amount of data must be processed prior to a deadline and the WAN connection to a central data center isn't sufficiently large enough to transfer the data to a central data center. The third is local deployment of Virtualized Network Functions.v

3.11 Model Checking of Threshold-based Fault-tolerant Distributed Algorithms

Igor Konnov (TU Wien, AT)

Joint work of Konnov, Igor; Gmeiner, Annu; Schmid, Ulrich; Veith, Helmut; Widder, Josef
Main reference A. John, I. Konnov, U. Schmid, H. Veith, J. Widder, "Parameterized model checking of fault-tolerant distributed algorithms by abstraction," in Proc. of the 13th Conf. on Formal Methods in Computer-Aided Design (FMCAD'13), pp. 201–209, IEEE, 2013.
URL http://dx.doi.org/10.1109/FMCAD.2013.6679411

Model checking of fault-tolerant distributed algorithms is challenging: the algorithms have multiple parameters that are restricted by arithmetic conditions, the number of processes and faults is parameterized, and the algorithm code is parameterized due to conditions involving counting the number of received messages (thresholds). We present our framework that allows us to model threshold-based fault-tolerant distributed algorithms and then efficiently model check them. To address parameterization, we introduced several model checking techniques for verification of such algorithms for all system sizes and all possible numbers of faults. We give an overview of these techniques and of our recent verification results.

3.12 Computation Distribution Networks and Distributed Clouds

Rick McGeer (HP Enterprise Services – Palo Alto, US)

The continuing performance increase of computation over communication has spurred the rise of many new network services over the past two decades: people are using the ubiquity of cheap, powerful computation to offer collaborative content distribution, transport proxies that offer high quality of service, in-network transcoding, adaptive routing, and so on. The efficacy of all of these services pales in comparison to the ability to arbitrarily site computation, particularly at the network edge: easily the most effective thing that one can do with a network is send a program over it. Specific use cases of computation distribution include in-situ data reduction, particularly for data with high-bandwidth sensors such as cameras, and high-bandwidth or latency-sensitive user interactions (real-time interactive simulation and distributed collaborative visualizations, for example). In this talk, I'll give some examples of computation distribution, and discuss a prototype of a Computation Distribution Network.

3.13 The Discovery initiative

Jonathan Pastor (INRIA Rennes – Bretagne Atlantique, FR)

Although the concept of Micro and Nano Data Centers (DCs) has been proposed to deliver more efficient as well as sustainable Utility Computing (UC) resources,the questions of where deploying and how federating thousands of such facilities are still far from being solved and the current trend of building larger and larger DCs in few strategic locations still prevails.The DISCOVERY initiative proposes to directly deploy the concept of Micro/NanoDCs upon the network backbones in order to benefit from existing network centers, starting from the core nodes of the backbone to the different network access points in charge of interconnecting public and private institutions.By such a mean, network and UC providers would be able to mutuality resources that are mandatory to operate network/data centers while delivering widely distributed UC platforms being able to better match the geographical dispersal of users.

3.14 Decomposing consistency

Marc Shapiro (UPMC – Paris, FR)

Joint work of Shapiro, Marc; Saeida-Ardekani, Masoud; Zawirski, Marek; Balegas, Valter; Najafzadeh, Mahsa; Gotsman, Alexey

There are many competing consistency models (serialisability, snapshot isolation, eventual consistency, etc.), all subtly different and hard to understand. This variety reflects a fundamental trade-off between fault tolerance, performance, and programmability. Furthermore, most papers describe consistency models in terms of acceptable histories, which is not very informative. The design choices are particularly vexing at large scale and in the presence of failure, for instance in geo-replicated or edge clouds. We believe that what programmers really care about is a consistency model's *properties.* We study two classes of properties: guarantees (i.e., what kind of application invariants are ensured automatically by a model) and scalability properties (i.e., opportunities for parallelism and implementation freedoms in a model). The properties are duals between the two classes; they are (mostly) orthogonal within a class. A particular composition of properties will characterize a model. We also study some abstract classes of guarantees (e.g., partial-order-type invariants, equivalance-type invariants, identical-observer guarantee) and opportunities (e.g. genuine partial replication) and their impact on the protocols and the models.

3.15 How to distribute cloud computing to the edge?

Hagen Woesner (BISDN GmbH – Berlin, DE)

URL http://www.fp7-unify.eu/

The talk introduces the *fat CPE* as opposed to other approaches of stripping down *virtualized CPEs* to a single tunnel endpoint. Some use cases motivate this choice. In the following, a universal node architecture is introduced (coming from EU project UNIFY). We end asking questions of how resources should be exposed: As network function forwarding graphs (NF-FG)?

3.16 Is the Cloud Your Data Center, Your Network, or Both?

Tim Wood (George Washington University – Washington, US)

Main reference J. Hwang, K. K. Ramakrishnan, T. Wood, "NetVM: High Performance and Flexible Networking using Virtualization on Commodity Platforms," in Proc. of the 11th USENIX Symp. on Networked System Design and Implementation (NSDI'14), pp. 445-458, USENIX Association, 2014.
URL https://www.usenix.org/conference/nsdi14/technical-sessions/presentation/hwang

Today's trends have been towards a handful of major cloud platforms composed of a small number of massive data centers. While this gives benefits from economy of scale, it may not be able to achieve the resiliency and performance of a more distributed model. This talk will discuss how software-based networks using SDN and NFV may help move us back towards a distributed cloud model by allowing software services to easily and efficiently run in the network itself. By pushing storage and computational capabilities into the network, we can perform computation precisely when and where it is needed, reducing latency and increasing fault tolerance.

4 Panel Discussions

Over the course of the seminar, we organized three panel discussions on the topics of (i) general aspects of distributed cloud computing, (ii) networking aspects of distributed cloud computing, and (iii) distributed systems and programmability aspects of distributed cloud computing. The panel discussions aimed to stimulate breakout sessions that identify research challenges.

4.1 Panel Discussion on Distributed Cloud Computing

Justin Cappos, Lars Eggert, Chip Elliott, Oliver Hohlfeld, and Rick McGeer

- Seattle: distributed cloud that is larger than Planet Lab
- What is a distributed cloud? It is an infrastructure with substantially distributed computation and storage resources. A CDN would not count as DC since they only serve traffic and provide no to limited possibilities of running compute jobs / programmability.
- It is all about latency. Latency makes nodes different from each other.
- What is the differences between traditional means of computation and distributed cloud computing? The cloud should simplify approaches.
- A lot of research is needed to meet performance criteria / SLAs in practice. There is no one button to be clicked to instantiate a cloud app that meets specified criteria.
- The motivation for cloud computing is driven by costs. Distributed cloud computing offers redundancy and reliability. In addition it provides instant fail-overs by having remote replicas that can be booted up in immediately increase of failures.
- Use Case 1: Educational games. There are reasons for not sharing game code with the end users: 1) prevent cheating / maintain control and 2) simplify portability: by only streaming video rendered by the game over the network, less platform need to be supported by game vendors.
- Use Case 2: Programs should be sent over the network rather than data.Programs are much smaller but typically generate large amounts of data. For example, detectors at the LHC at Cern can generate 3M events in a short amount of time. Since not all of these events are of interest, the network utilization can be reduced by moving a filter program next to the sensor.
- Argument challenged: Big data applications like LHC will always rely on dedicated infrastructures. Ordinary users will not run compute job on LHC cloud nodes and will demand different infrastructures.
- Sending VMs to storage systems is challenging since the CPU capacity is limited and will be occupied by compute jobs. It is better to move the data to dedicated compute

clusters for processing when the data is hot. The processing pipeline should focus on hot data rather than cold data.
- Privacy vs. performance
- Privacy concerns drive migrations to private clouds running Open Stack. Another approach relies on moving storage devices owned by the data owners next to big data centers (e.g., Amazon) for fast-path access to compute clusters.
- What are the programming models for distributed cloud computing? (other than simply placing a VM)
- Infrastructure management / DHT centralized lookup systems
- Support for desktop orchestration. It should be possible to add a distributed file system as fuse model to be run at desktops.

Identified discussion topics:
1. What is a distributed cloud?
2. Programmability models
3. Scalability and repeatability of experiments
4. How can academia compute with big companies?
5. Service composition / Decomposition – what are the primitives for infrastructure management?
6. What are the applications using distributed cloud computing?
7. Is there an educational use for distributed cloud computing?

4.2 Panel Discussion on Networking Aspects of Distributed Cloud Computing

Mark Berman, Lars Eggert, Oliver Hohlfeld, James Kempf, and Hagen Wösner

- Discussion on the Open Stack networking architecture
- Open Stack runs the VM placement algorithm first and then performs the wiring of the placed VMs. Emulab uses simulated annealing for its optimization. Can such an approach be applied? How do VM images get copied over the network?
- Shipping VM images in principle is a bad idea. They should be kept local / cached.
- We use dedicated fibers are used to ship images. It can be used but does not represent the general case.

Use cases:
- Future robot control software generates massive amounts of data that need to be processed in a local cloud since the robots are not powerful enough.
- Service providers would like to move VMs to users' homes for debugging. Such a use case was once built on top of Seattle and is also discussed in the ETSI NFV use cases.
- Home gateways should have higher compute powers in order to run VMs.

Identified discussion questions:
1. Relationship between clouds and CPE's? Does the cloud land in my house or is my house outsourced to some cloud?
2. Joint placement of CFNet
3. Clouds in mobile networks
4. Cloud to WAN: next steps

4.3 Panel Discussion on Aspects of Distributed Systems and Programmability

Annette Bieniusa, Patrick Eugster, Oliver Hohlfeld, Marc Shapiro, and Tim Wood

Which programming primitives are needed for distributed cloud computing?

- Virtual machine is an execution model but not a programming model.
- We need a more fine grained understanding of what a programmer intends to do with resources provided by distributed clouds. E.g., it might be worth to migrate single users of a VM rather than the entire VM.
- Which primitives are needed for distributed state
- There can be different kinds of state that require different consistency models. Thus, a library providing support for a large variety of such models is needed.
- At which level is such a library envisioned? At a service composition or at a bit level? A generic model will not provide good performance at all layers, i.e., there should also be specialized models be available.
- In distributed systems, we didn't end up needing as much consistency as we initially thought.
- Programming languages failed in the distributed world as there is no one size fits it all in distributed systems. Different requirements need to be supported. Why is distributed cloud computing different from classical work in distributed systems?
- Different application demands
- Trend towards domain specific languages Orthogonal to these 3 areas, there is *latency* and *faults*.
- Variations in performance will be a big challenge in distributed cloud computing.

What is the appeal for the educational use of a distributed cloud platform such as Seattle?

- Seattle simplifies experimentation with a distributed networked systems. It is very appealing to school students to test their approaches by running at remote computers, e.g., phones located in China.

The following questions were identified by the panel:

1. Fault tolerance
2. Latency requirements
3. If one size doesn't fits it all, how many sizes do we need?
4. Configuration management
5. Data quality
6. Requirements and models of different applications run on distributed clouds
7. Automated decision support

5 Working Groups Identifying Research Challenges

We used questions identified in the panel discussions to form and stimulate working groups. The formed working groups then identified potential research questions in their respective areas.

5.1 Research Challenges in Distributed Clouds

This breakout session concerned identifying research challenges that arise in distributed clouds. We identified five research questions out of which one is dedicated to programming language techniques and one to networking aspects.

- In multi-domain distributed clouds, different parts of the cloud will be owned and operated by different organizations. How will resources be described and obtained? Will there be brokers? How does multi-domain management works?
- If parts of the cloud will be battery powered, what impact will this have on the architecture in terms of reliability, consistency, etc.?
- What are the 3 to 4 driving use cases for distributed cloud computing?
- Programming Languages: When is consistency needed and to what degree? When is BFT needed and to what degree? How frequently will apps need these techniques and how does one know they are worth the complexity?
- Distributed Systems: How does SDN relate to distributed cloud computing,if at all?

5.2 Distributed Systems Research Challenges for Distributed Cloud

This breakout session concerned identifying research challenges that arise in distributed systems as a driver for distributed clouds. We identified three research questions for distributed systems, and two for the other two breakout groups (clouds and programming languages).

- (Distributed Systems) How about performing a threat analysis on some consistency mechanisms to determine possible security issues?
- (Distributed Systems) How do you write control algorithms for controlling the distributed cloud that are salable?
- (Distributed Systems)In causal consistency, what would the effect be of replacing the deterministic causal graph with a probabilistic causal graph?
- (Programming Languages) How to provide good abstractions in the face of heterogeneous underlying resources (i.e. not making the abstractions least common denominator and not providing so many parameters that the abstractions become too complex)?
- (Clouds) What are the trust models for distributed cloud and does distributed cloud make security easier or harder?

5.3 Programmability Group

- How do we deploy and monitor apps running on DCC?
- How is programmability constrained by the infrastructure?
- Which primitives do we need to express what we want?
  - Primitives for describing the placement of components- Primitives for configuring the non-functional requirements
  - Primitives for configuring the functional requirements- Low-level vs. High-level primitives
  - Dynamic environment discovery
  - Primitives for SLA
- What are the relevant applications? Identify major classes of applications. Suggestions:
  - Big data analytics
  - Services (control loop, drive my car, do face matching)

  - Games MMOG- Latency-sensitive applications
  - Web servers

  What are their requirements in terms of programmability? Note: all these kinds of applications define what a DCC is.
- How do we educate developers?
  - Focus on testing
  - Focus on performance evaluation
- Question for the cloud group: How do you build an infrastructure allowing the deployment of computations and data wherever needed?
- Question for the distributed system group: How do we address the placement issue? → we might want these two components to be far away. Or to be on the same computer. Or on one particular resource.

Participants

- Mark Berman
BBN Technologies –
Cambridge, US
- Annette Bieniusa
TU Kaiserslautern, DE
- Justin Cappos
New York University, US
- Yvonne Coady
University of Victoria, CA
- Lars Eggert
NetApp Deutschland GmbH –
Kirchheim, DE
- Johan Eker
Lund University / Ericsson
Research, SE
- Chip Elliott
BBN Technologies –
Cambridge, US
- Erik Elmroth
University of Umeå, SE
- Patrick Eugster
Purdue University, US
- Hannes Hartenstein
KIT – Karlsruher Institut für
Technologie, DE
- Oliver Hohlfeld
RWTH Aachen, DE
- James Kempf
Ericsson – San Jose, US
- Igor Konnov
TU Wien, AT
- Rick McGeer
HP Enterprise Services – Palo
Alto, US
- Christine Morin
INRIA Rennes – Bretagne
Atlantique, FR
- Jörg Ott
Aalto University, FI
- Jonathan Pastor
INRIA Rennes – Bretagne
Atlantique, FR
- Vivien Quéma
INP – Grenoble, FR
- Stefan Schmid
TU Berlin, DE
- Marc Shapiro
UPMC – Paris, FR
- Hagen Wösner
BISDN GmbH – Berlin, DE
- Tim Wood
George Washington University –
Washington, US