Operating Complex IT-Systems Seminar – Collection of Topics

Winter term 2015/16

This document lists the seminar topics selected by our research stuff for this term's bachelor and master seminars. Each student is encouraged to choose at least 3 desired topics form the provided set. Each topics includes an abstract, references and a recommendation regarding the applicability for bachelor or master students (or possibly both). The references are to be used as a foundation for the literature review, which is required to prepare the slides and the scientific paper. Note that it is not sufficient to rely on the provided references only. If you have trouble accessing the references (e.g. through ieeexplore) use a workstation that is either physically or virtually (e.g. by VPN) connected to the university network. You may additionally try to locate a free-access version of the reference using the scholar search engine offered by Google.

Further notes:

- Please make sure not to miss the service talks on scientific writing and presentation on 12th of November (see website for details)
- Final assignment of the topics takes place during the kickoff meetings on 28th of October. Participation is mandatory. Students not showing up unexcused automatically cancel their registration.
- Please elect more than one topic of interest. We will try to resolve conflicts, but we can't guarantee that each student will be assigned his favorite topic.

Topics sorted by area of research

1 Distributed Systems Theory

Topic 1: Virtual Time
Abstract:
Virtual time is a new paradigm for organizing and synchronizing distributed systems which can be applied to such problems as distributed discrete event simulation and distributed database concurrency control. Virtual time provides a flexible abstraction of real time in much the same way that virtual memory provides an abstraction of real memory. It is implemented using the Time

Warp mechanism, a synchronization protocol distinguished by its reliance on lookahead-rollback, and by its implementation of rollback via antimessages.	
References:	
http://dl.acm.org/citation.cfm?id=3988	
Applicable for BSc: no	**Applicable for** MSc: yes
Further Notes: no	

Topic 2: Optimistic Concurrency Control	
Abstract:	
Most approaches to concurrency control in database systems rely on locking of data objects as a control mechanism. In this paper, two families of nonlocking concurrency controls are presented. The methods used are “optimistic” in the sense that they rely mainly on transaction backup as a control mechanism, “hoping” that conflicts between transactions will not occur. Applications for which these methods should be more efficient than locking are discussed.	
References:	
[1] http://dl.acm.org/citation.cfm?id=319567 [2] http://redis.io/topics/transactions	
Applicable for BSc: no	**Applicable for** MSc: yes
Further Notes: no	

Topic 3: Software Transactional Memory	
Abstract:	
In computer science, software transactional memory (STM) is a concurrency control mechanism analogous to database transactions for controlling access to shared memory in concurrent computing. It is an alternative to lock-based synchronization. STM is strategy implemented in software, rather than as a hardware component. A transaction in this context occurs when a piece of code executes a series of reads and writes to shared memory. These reads and writes logically occur at a single instant in time; intermediate states are not visible to other (successful) transactions.	
References:	
[1] http://groups.csail.mit.edu/tds/papers/Shavit/ShavitTouitou.pdf [2] http://dl.acm.org/citation.cfm?id=872048 [3] http://blog.enfranchisedmind.com/2009/01/the-problem-with-stm-your-languages-still-suck/	
Applicable for BSc: no	**Applicable for** MSc: yes
Further Notes: no	

Topic 4: Differential Synchronization	
Abstract:	
Differential Synchronization (DS) method is for keeping documents synchronized. The key feature of DS is that it is simple and well suited for use in both novel and existing state-based applications without requiring application redesign. DS uses deltas to make efficient use of bandwidth, and is fault-tolerant, allowing copies to converge in spite of occasional errors. We consider practical implementation of DS and describe some techniques to improve its performance in a browser environment.	
References:	
[1] https://neil.fraser.name/writing/sync/eng047-fraser.pdf [2] https://spring.io/blog/2014/10/22/introducing-spring-sync	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

Topic 5: Geo Replication	
Abstract:	
Geo-replication systems are designed to improve the distribution of data across geographically distributed data networks. This is intended to improve the response time for applications such as web portals. Geo-replication can be achieved using software, hardware or a combination of the two. Online services distribute and replicate state across geographically diverse data centers and direct user requests to the closest or least loaded site. While effectively ensuring low latency responses, this approach is at odds with maintaining cross-site consistency.	
References:	
[1] https://www.usenix.org/system/files/conference/osdi12/osdi12-final-162.pdf [2] https://www.usenix.org/system/files/conference/osdi14/osdi14-paper-ardekani.pdf	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

Topic 6: CoRAL – Reliable Web Services	
Abstract:	
Making stateful web services reliable requires elaborate cross-layer techniques. The fault tolerance scheme CoRAL (Connection Replication and Application-level Logging) actively replicates the state of a TCP connection and additionally logs HTTP requests/replies to enable fast failover to a warm-standby server.	
References:	
http://web.cs.ucla.edu/csd/research/labs/csl/projects/coral/ http://web.cs.ucla.edu/~tamir/papers/pdcs03.pdf http://millennium.cs.ucla.edu/~tamir/papers/coral_jss09.pdf	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

Topic 7: Distributed State Machines	
Abstract:	
Maintaining consistent application state is an important issue when implementing replicated network services. Paxos is a widely used algorithm for implementing a Distributed State Machine which allows a number of service replicas to maintain consistency. Paxos has been extended and improved many times since Lamports original description of the algorithm.	
References:	
Paxos http://dl.acm.org/citation.cfm?doid=279227.279229 http://www.ux.uis.no/~meling/papers/2013-paxostutorial-opodis.pdf ***NetPaxos*** NetPaxos is an extension to Paxos optimizing it for the use in modern SDN-capable switches. http://perso.uclouvain.be/marco.canini/papers/netpaxos.sosr15.pdf ***Raft*** Raft is a novel consensus algorithm inspired by Paxos designed to be more understandable for students of dependable systems while providing the same consistency guarantees and performance as Paxos. http://www.eecs.harvard.edu/cs261/papers/ongaro14.pdf	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: Besides the original Paxos algorithm, this topic contains 2 sub-topics dealing with extended approaches. Students can decide whether to present one topic in detail or focus on a comparison. Furthermore, this topic may be assigned to up to three students as well.	

2 Large Scale Cluster Infrastructures & Management

Topic 8: Publish Subscribe with Apache Kafka	
Abstract:	
In modern distributed systems the communication between components often relies on messaging. Kafka is a distributed messaging system that was developed for collecting and delivering high volumes of log data with low latency. A few unconventional yet practical design choices in Kafka make it efficient and scalable with performance superior to popular alternatives.	
References:	
Primary literature: Kafka: a Distributed Messaging System for Log Processing http://notes.stephenholiday.com/Kafka.pdf Secondary literature: https://www.rabbitmq.com/	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

Topic 9: Quasar: Resource-Efficient and QoS-Aware Cluster Management	
Abstract:	
Quasar is a cluster management system that increases resource utilization while providing consistently high application performance. Instead of resource reservations, which lead to underutilization as users do not necessarily understand workload dynamics and physical resource requirements of complex codebases, users express performance constraints for workloads, letting Quasar determine the right amount of resources to meet these constraints. Quasar uses classification techniques to quickly and accurately determine the impact of the amount of resources (scale-out and scale-up), type of resources, and interference on performance for each workload and dataset.	
References:	
https://web.stanford.edu/group/mast/cgi-bin/drupal/content/quasar-resource-efficient-and-qos-aware-cluster-management https://www.youtube.com/watch?v=YpmElyi94AA	
Applicable for BSc: no	**Applicable for** MSc: yes
Further Notes: no	

Topic 10: Faulttolerance in MapReduce	
Abstract:	
MapReduce is an often used Parallel programming Framework. It enables the user to run jobs within the cloud on many machines. The usage of a lot of working nodes, leads to an increase of failure probability. Therefore it is necessary to work on fault tolerance for MapReduce Systems. There are several ideas how to improve fault tolerance in MapReduce:	
References:	
Improving MapReduce fault tolerance in the cloud http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5470865& Towards self-caring mapreduce: Proactively reducing fault-induced execution-time penalties http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5999808 HPCLOUD: A novel fault tolerant architectural model for hierarchical MapReduce http://ieeexplore.ieee.org/mobile/Abstract.jsp?arnumber=6206822	
Applicable for BSc: no	**Applicable for** MSc: yes

Further Notes: no

Topic 11: Comparison of Distributed File systems for Parallel Execution	
Abstract:	
Running jobs in the cloud in parallel raises the needs of storing data efficiently. Usually parallel execution Frameworks like MapReduce use distributed File systems. For example HDFS, Google FS, CODA, GlusterFS... But which one should be used, in which case?	
References:	
HDFS http://pristinespringsangus.com/hadoop/docs/hdfs_design.pdf GFS http://dl.acm.org/citation.cfm?id=945450 CODA http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=54838	
Applicable for BSc: yes	**Applicable for** MSc: no
Further Notes: This topic is related to an available/vacant bachelor thesis. In case you wish to use your seminar results as a foundation for a BA thesis literature review, contact Mareike Höger for further details.	

Topic 12: Twitter Heron: Stream Processing at Scale	
Abstract:	
Twitter Heron was developed to address the limitations Twitter identified with its initial distributed stream processor Apache Storm. Heron is now the standard engine in production at Twitter and is claimed to surpass Storm in all important categories: it scales better, has better debug-ability, has better performance, and is easier to manage in shared cluster setups.	
References:	
http://dl.acm.org/citation.cfm?doid=2723372.2742788 https://blog.twitter.com/2015/flying-faster-with-twitter-heron http://dl.acm.org/citation.cfm?id=2595641	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

Topic 13: Google's Dataflow: Balancing Correctness, Latency, and Cost in Massive-Scale, Unbounded, Out-of-Order Data Processing	
Abstract:	
Google's Dataflow engine can not only process and aggregate data from continuous event streams, but also provides mechanisms to handle events that arrive late. By defining reactions on late events and trigger policies, users make explicit tradeoffs between the correctness, latency, and cost of processing large-scale continuous inputs.	
References:	
http://dl.acm.org/citation.cfm?id=2536229, http://www.vldb.org/pvldb/vol8/p1792-Akidau.pdf https://cloud.google.com/dataflow/	
Applicable for BSc: no	**Applicable for** MSc: yes
Further Notes: no	

Topic 14: Large-scale cluster management at Google with Borg
Abstract:

Google's Borg system is a cluster manager that runs hundreds of thousands of jobs across a number of clusters each with up to tens of thousands of machines. It achieves high utilization by combining admission control, efficient task-packing, over-commitment, and machine sharing with process-level performance isolation. Borg simplifies life for its users by offering a declarative job specification language, name service integration, real-time job monitoring, and tools to analyze and simulate system behavior
References:
research.google.com/pubs/archive/43438.pdf http://research.microsoft.com/pubs/232978/osdi14-paper-boutin.pdf

Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

3 Cloud Infrastructures & Software Defined Networking

Topic 15: XMPP as an extensible Message Oriented Middleware protocol	
Abstract:	
XMPP is a message-oriented communication protocol that is based on TCP and enables the near-real-time exchange of small pieces of structured data which are called XML stanzas. XMPP provides among authentication and data protection many features which make this protocol and its extensions an ideal candidate for an extensible middleware or Message Oriented Middleware protocol.	
References:	
http://xmpp.org/ http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6924206	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

Topic 16: APIs for Cloud Interoperability	
Abstract:	
Clouds are often provided within a data centers of a single institution. Here a cloud middleware manages the access to virtualized resources through a cloud interface that can be a web service Application Programming Interface (API) or a more user-friendly web application. Because of the novelty of the cloud paradigm, many companies started working on their own proprietary standards for virtual machine configurations, their associated file formats, the application packaging and its deployment through proprietary interfaces. These lacks of interoperability had been solved by standards like the Open Virtualization Format (OVF), the Open Cloud Computing Interface (OCCI), and the Topology and Orchestration Specification for Cloud Applications (TOSCA).	
References:	
http://www.cloud-council.org/CSCC-Cloud-Interoperability-and-Portability.pdf http://occi-wg.org/ http://jclouds.apache.org/	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

Topic 17: Network Monitoring Systems
Abstract:
OpenNMS and Nagios are open source network monitoring and management solutions and are delivered with different features like event management and notification, service monitoring, discovery and provisioning, graphical interfaces, etc. The high customizability, a sufficient amount of graphical components and the accessibility by web-interfaces make that technologies an

appropriate candidate for monitoring an application infrastructure which manages IoT devices in a self-configurable and self-sustaining manner.
References:
==== OpenNMS ======= http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=F3FAF1B59347CF187C6075537FB60C06?doi=10.1.1.128.2742&rep=rep1&type=pdf http://bth.diva-portal.org/smash/get/diva2:830767/FULLTEXT01.pdf ==== Nagios ======= http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6304293 http://203.185.67.136/inmsweb/paper/Apricot_nagios20091130-final.pdf

Applicable for BSc: yes	**Applicable for** MSc: no
Further Notes: no	

4 Internet of Things & Machine-2-Machine Communication

Topic 18: Future Internet Architectures – The FI-WARE Project	
Abstract:	
The FI-WARE platform targets the creation of applications for the Future Internet including application domains such as smart cities, transportation, logistics. The core is a set of platform components (i.e. APIs), called Generic Enables, that allow developers to easily implement smart applications through plugging enablers together and deriving domain specific specializations from them. Furthermore, the Fi-WARE platform consists of a federated development Cloud spread around multiple locations of Europe. Each FI-WARE node provides the mandatory resources and APIs (Generic Enablers) and allows developers to implement platform- and location independent Future Internet applications. The objective of this topic is to introduce the FI-WARE ecosystem and platform with focus on the Cloud platform and the Generic Enablers.	
References:	
https://www.fiware.org/ https://www.fiware.org/developers-entrepreneurs-temp1/	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

Topic 19: Participatory Sensing Systems	
Abstract:	
The idea behind participatory sensing is to take advantage of the various sensing capacities owned by individuals or communities. Instead of deploying sensing nodes dedicated to a particular application scenario, applications may utilize the sensors already integrated into the smart phones most of us use every day. The objective of this topic is to introduce architectures for participatory sensing and discuss related issues such as security and privacy.	
References:	
http://escholarship.org/uc/item/19h777qd https://www.emilianodc.com/PEPSI/	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

Topic 20: Architecting the Internet of Things
Abstract:
The Internet of Things (IoT) recently gains attention as a generic term for solutions from various application domains. Examples are smart homes, smart cities, factory automation, e-Health,

transportation, and logistics. However, most specific solutions rely on vendor- or application domain dependent architectures. These architectures barely interact and are usually deployed on top of isolated and heterogeneous technical infrastructures. As a consequence, applications cannot unleash the full potential of the various sensor nodes and embedded systems deployed. As a mitigation, various standardization initiatives currently investigate in providing a common understanding of IoT architectures.

References:

*** ETSI – oneM2M ***
http://www.etsi.org/about/what-we-do/global-collaboration/onem2m
*** IIC ***
http://www.iiconsortium.org/IIRA.htm
https://www.youtube.com/watch?v=tcMH9NV0YgQ
*** IoT-A ***
http://www.iot-a.eu/public
http://link.springer.com/book/10.1007%2F978-3-642-40403-0

Applicable for BSc: yes	**Applicable for** MSc: yes

Further Notes: This topic is subject to teamwork. It can be assigned to 2-4 students. The main objective is to provide a meaningful comparison of the presented architectures. Each student has to get in touch with at least one of the listed architectures.

Topic 21: VSN - Virtual Sensor Networks

Abstract:

Motivated by the increasing capabilities of the nodes participating in WSNs, Cloud characteristics like pooling and allocating a subset of the resources to certain tasks on-demand are applied to sensor networks. Virtual Sensor Networks (VSNs) assume that the overall set of nodes in a WSN can be virtually grouped into subsets and dedicated to specific applications or tasks. Unlike traditional WSNs, where usually all nodes perform similar tasks, a subset of nodes can be allocated for a given time period to perform specific tasks or react to the current environment. A sensor network deployed in an area recovering from a disaster may, for instance, observe specific events in a certain region. Using VSN, nodes close to that region could be grouped and further investigate the situation by executing specific tasks.

References:

http://www.mdpi.com/1424-8220/12/2/2175
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4151668

Applicable for BSc: yes	**Applicable for** MSc: yes

Further Notes: no

Topic 22: Sensor Virtualization

Abstract:

Sensor virtualization aims at overcoming the resource constraints of traditional WSN by integrating Cloud resources and providing access to the physical sensors to multiple users. Sensor Virtualization approaches allow simplifying the management of and access to the sensors using standard Cloud interfaces and applications that hide the actual location and the diversity of the sensors from the consumers (e.g. abstraction from technical details, focus on the semantics, grouping of functional identical sensors).

References:

http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5678060
http://www.hindawi.com/journals/ijdsn/2013/917923/

Applicable for BSc: yes	**Applicable for** MSc: yes

Further Notes: no

Topic 23: TEDS – Transducer Electronic Data Sheets	
Abstract:	
IEEE 1451 is a collection of standards that aims at providing uniform interfaces to interact with sensors or actuators (transducer) independently of the underlying communication technology. The core element of the standard collection is the definition of the Transducer Electronic Data Sheet (TEDS), which contains the information required by a measurement system to interact with a transducer (e.g. ID, calibration and correction data) and therefore allows integrating the transducer in a plug and play manner. The TEDS can be stored within the memory of the transducer or as a separate file downloadable from the internet if legacy transducers without memory need to be integrated. Compared to other device discovery and description technologies like UPnP, TEDS is more suitable for highly resource constrained devices.	
References:	
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5668466 http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4483728	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

Topic 24: CoAP – Constrained Application Protocol	
Abstract:	
As the name suggests, the Constrained Application Protocol (CoAP) is an application level protocol dedicated to resource constrained devices and M2M or IoT applications. It was designed according to RESTful principles in order to be easily mapped to HTTP while keeping focus on low overhead and simplicity. In contrast to web service approaches like the Device Profile for Web Services (DPWS), CoAP uses a binary header format. Similar to DPWS, it can operate on top of 6LoWPAN. The objective of this topic is to introduce the CoAP protocol and its use cases within the IoT domain.	
References:	
https://tools.ietf.org/html/rfc7252 http://www.mdpi.com/2224-2708/2/2/235/htm http://coap.technology/ http://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=Coap:%20An%20application%20protocol%20for%20billions%20of%20tiny%20internet%20nodes	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

5 Big Data Analytics & Visualization

Topic 25: Visualization of Highly-Connected Big Data	
Abstract:	
Highly-connected big data, modeled as a graph, typically consists of trillions of edges. Such large amounts of data can neither be understood nor displayed at once. Therefore novel visualization and vertex/edge organization approaches must be developed.	
References:	
https://www.dropbox.com/s/dyhb2fodabfnt29/ma13.pdf?dl=0 https://www.dropbox.com/s/92twlvzxc1va2dx/05290699.pdf?dl=0	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: Upon request, this topic may be split into two separate ones.	

Topic 26: Exploiting Internet Knowledge Bases for Concept Detection in Big Data
Abstract:

In order to detect concepts in big data, a model of the real world needs to exist as a basis. To create this model, current approaches exploit the knowledge freely available on the internet, such as Wikipedia or Wordnet.
References:
https://www.dropbox.com/s/fov86nqszu3zqcz/1503.01655.pdf?dl=0 https://www.dropbox.com/s/awr6bd1xytquxyf/aij_2012_navigli_ponzetto.pdf?dl=0
Applicable for BSc: yes | **Applicable for** MSc: yes
Further Notes: no

Topic 27: Finding Relationships between Concepts in Big Data	
Abstract:	
When modeling big data as a graph, the edge information can be exploited in order to detect relationships between concepts. However, this is a tricky task given the extremely low diameter and, at the same time, extremely high amount of edges/vertices.	
References:	
https://www.dropbox.com/s/bgfukl2vc86j47i/93660561.pdf?dl=0 https://www.dropbox.com/s/dzabm5kvfnp1s7i/tacl_2014_babelfy.pdf?dl=0	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

Topic 28: Concept Detection in Big Data	
Abstract:	
One of the biggest challenges in computer science is to answer the question, what big data is about. The answer can be given in terms of recognized concepts, which do not necessarily have to appear as symbols in the data itself.	
References:	
https://www.dropbox.com/s/ojwtsd3rsoh9kac/cambell_et_al_j_of_advertising_2011.pdf?dl=0 https://www.dropbox.com/s/2wl9ie8ny4rfon9/305-869-1-pb.pdf?dl=0	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

Topic 29: Entailment Graphs	
Abstract:	
Entailment graphs are used in order to model and derive implications within big data. The knowledge produced in this way can help to support approaches extracting meaning from information.	
References:	
http://csli-lilt.stanford.edu/ojs/index.php/LiLT/article/viewFile/5/4 http://www.dfki.de/~neumann/publications/new-ps/TDS2014.pdf	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

Topic 30: Feature Extraction from Big Data
Abstract:
In recent years great progress has been made in the field of feature extraction from big data through supervised methods. The high quality of the extracted features creates the potential to vastly improve the accuracy of approaches which extract meaning from big data without supervision.
References:

http://arxiv.org/pdf/1412.2306.pdf http://www.cs.cmu.edu/~ninamf/courses/601sp15/slides/23_nell_4-13-2015.pdf	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

6 Miscellaneous Topics

Topic 31: Java 8 Stream API	
Abstract:	
Version 8 of the Java Platform introduced features that allow developers to leverage the advantages of a functional programming based syntax (lambda expressions). Furthermore, the new Stream API allows to process data in a declarative way and provides features to utilize multi-core architectures without having to write multithreaded code. The objective of this topic is to introduce the fundamental, functional programming inspired, concepts of the Java 8 stream API and lambda expressions to the fellow students.	
References:	
http://www.oracle.com/technetwork/articles/java/ma14-java-se-8-streams-2177646.html http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/Lambda-QuickStart/index.html http://www4.fh-swf.de/media/downloads/fbtbw/download_8/devries_1/Funktionen-Streams.pdf (german)	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

Topic 32: Micro-Services	
Abstract:	
The term Micro-Services is related to a style or pattern in software architecture. Single applications are developed as a suite of small services. These services interact with each other using inter process communication and language agnostic APIs. The objective of this topic is to introduce the micro-services architecture style and distinguish it from related styles such as SOA.	
References:	
http://martinfowler.com/articles/microservices.html http://nirmata.com/2015/02/microservices-five-architectural-constraints/ http://injoit.org/index.php/j1/article/view/139	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

Topic 33: Netflix Open Source Software Center	
Abstract:	
Neflix uses various Open Source Software (OSS) to serve the real-time load to over 62 million users. Major challenges are the maintenance of data persistence, build environments, security, big data analytics, reliability and performance of the service at massive scale. Netflix bundles the used software and provide the collection as Netflix Open Source Software Center.	
References:	
https://netflix.github.io/ http://techblog.netflix.com/	
Applicable for BSc: yes	**Applicable for** MSc: yes

Further Notes: This topic is related to the Micro-Services topic. The presentation should be prepared with regard to the presence of the more general Micro-Services presentation in the same seminar. Thus, if no Micro-Services talk will be given by a fellow student, a short introduction to the topic may be eligible. Otherwise, the a Micro-services introduction may be skipped.

Topic 34: Distributed Module Systems – R-OSGi	
Abstract:	
OSGi is a specification of a platform-independent (Java based) framework for modularized development of applications and services based on the principle of component-based software engineering [68]. A component, sometimes also referred to as a module, is a package of software that encapsulates related functions and data and provides these as services to other components using well defined interfaces. A component provides the implementation of a certain set of system processes defined by an interface, which is accessible by other components. According to the modularity of a system, an important capability of components is their exchangeability. OSGi allows updating or replacing a component at runtime without breaking the system. Thus, OSGi provides a dynamic service execution environment that is able to deploy and wire services in shape of components at runtime. The objective of this topic is to introduce concepts such as R-OSGi and distributed OSGi. These concepts aim to extend the so far centralized OSGi approach with features for distributed module management.	
References:	
http://people.inf.ethz.ch/troscoe/pubs/middleware07-rosgi.pdf https://cxf.apache.org/distributed-osgi.html	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

Topic 35: OAuth2.0 and OpenID-Connect	
Abstract:	
OAuth2.0 is an authorization protocol that follows an interesting approach with regard to the peer to peer resource sharing. When using Cloud services, we often face situations where we grant access to a service to resources hosted by another service (e.g. a social network service that integrates with an e-mail service). This means the resource requesting client (i.e. the third party operating the Cloud service) differs from the resource owner (i.e. the user). Thus, the resource owner usually has to share its credentials with the third party, which introduces serious obstacles (access cannot be revoked without revoking access of all third parties, third party usually stores the user's credentials, validity of access is difficult to manage). OAuth separates the client from the resource owner and issues. OpenID-Connect provides an authentication layer on top of the OAuth2.0 protocol. The objective of this topic is to introduce the basic concepts of Oauth2.0 and OpenID-Connect with regard to operating secure Cloud services.	
References:	
http://oauth.net/2/ http://openid.net/connect/ http://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=Inside%20the%20identity%20management%20game	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	

Topic 36: Metrics for Evaluating Recommender systems
Abstract:

Recommender systems are now popular both commercially and in the research community. Due to the wide variety of application domains, different approaches and metrics have been adopted for their evaluation. It is important from the research perspective, as well as from a practical view, to be able to decide on an algorithm that matches the domain and the task of interest.	
References:	
http://dl.acm.org/citation.cfm?doid=1577069.1755883 http://www.ict.swin.edu.au/personal/jgrundy/papers/rsse2014.pdf	
Applicable for BSc: yes	**Applicable for** MSc: yes
Further Notes: no	