CS5412 study guide

This is purely a list of topics from each lecture that could arise on the prelim (which will be short-answer questions, none requiring more than 2 or 3 sentences to answer). "Could" doesn't mean "will", obviously. But you should be comfortable with these topics.

Some topics have associated examples of questions you would want to practice on. If you can comfortably answer those, you would be ready for those kinds of questions on the prelim (or in a job interview!)

	Topics
Lecture 1	History of cloud computing. How were cloud systems created prior to ~2005? What broke and why? List a number of trends that all forced changes. How is a single web server that builds pages different from a tier-one web server that relies on a collection of micro-services (μ-services)? List a number of things that would still occur in a tier-one web server List some things that would be done in a μ-service for a website like Amazon In 2 or 3 lines define a message bus and its role. In 2 or 3 lines define a message queue and explain its role. Give 1 example of something we would probably do with a message bus, not a queue. Give 1 example of something we would probably do with a message queue, not a bus. Why not use just one style of messaging for everything? List some technical concerns. In 1 or 2 lines each, define the two styles of virtualization. Give 1 concrete example of how a container differs from a true VM. How do lightweight threads, heavyweight threads with their own cores, standard Linux processes and containers differ? After all, we talk about running programs with all of them, so they obviously aren't the same, yet they all "run code"! What are some ways that the computer hardware shapes efficiency for these different options for running programs? Define the C, A and P in CAP. Now state Brewer's CAP conjecture. In class we heard that CAP is sometimes viewed as a theorem, but sometimes people disagree with that assertion. In 2 or 3 lines, explain the controversy. What does it even mean to "relax C in order to offer AP"? Be concrete: exactly what would a system that relaxes C be doing? In what sense is C being relaxed, relative to a system in which C is not relaxed? Are CA and CP meaningful? If so, what would they mean? Give an example of a situation in a cloud edge IoT context where you can relax C. Now give another example where you can't relax C. (Make sure to be clear why C didn't matter in the first case but does matter for the second one. This will require you to relate the broad meaning of C to your specific use case and the specific meaning C would have in your setting.)
Lecture 2	Describe some patterns in which load fluctuation is seen in the cloud. Define elasticity and relate elasticity to load fluctuation. Jim Gray's core point: locking of the kind seen in transactional databases will cause increasing rates of deadlock (tied to the birthday paradox) and those transactions will abort and retry (hence get executed multiple times). Thus adding servers or adding

	more transactions causes a rapid collapse of performance (polynomial in the number of servers or transactions). The exponents are surprisingly big, too: s^3 and t^5. The bottom line is that scaling, other than by adding a passive replica to take over after failures (which only needs mirroring and is simple) will not be successful. Jim Gray argues that this forces a sharded model – we have no choice. What is state machine replication in a sharded replication system? Answer: every member sees the same requests in the same order, hence can stay in the same state, with no need for locks or transactions or abort/retry (Jim's analysis won't apply here). Plus, all the replicas can do read-only work, allowing us to spread read loads. What architectural requirements arise as a consequence of needing elasticity? Why are large cloud vendors forced to build systems with large numbers of point-of-presence subsystems and larger cloud datacenters, rather than just using cloud data-centers for everything? What kinds of μ-services would you find in a point-of-presence subsystem? What might reside in the full cloud datacenter but probably not be found in the point-of-presence subsystem? What is a "blob"? Define the term and give two examples of things that fall into this category. What are the major "places" where Facebook blob caching occurs? What role does Akamai's CDN play in the Facebook blob-serving structure? Where does image resizing occur in the Facebook infrastructure? Give an example where a cache for some part of Facebook sees a different pattern of load than some other cache, purely as a consequence of which part of the Facebook hierarchy the cache is serving. Do all of the blobs managed by Facebook have identical popularity? Explain why or why not. Why was it important for the S4LRU policy to be implemented on flash storage units? What are some limitations associated with flash storage? How did RIPQ turn out to be important for supporting S4LRU over flash?
Lecture 3	What is a function server? Define "function" in this context, and tell us what causes the function to run, how the function can specify that it will connect to μ-services, and what sort of code one writes. Suppose that we wanted to use a function server to control drones that are surveying the health of a field of some crop like corn. Would you use a function to compute the search pattern at the start of the task? Why? What kinds of events might the drone generate? In Azure IoT the actual connection to a sensor is made by the Azure IoT Hub. What would constitute an "event"? Suppose the drone has reached the end of the row and needs new instructions. Walk us through the steps by which it obtains instructions on what to do next. Define "stateless" and give an example of a function that is stateless, and an example of a function that has state. If a function server is stateless, does this imply that it has no variables? Where could a stateless function store state if it needed to? How could it update the stored state? Would locking be required? What is a state machine? What is a digraph (directed graph)? Why is it common to use a digraph to describe a state machine?

	Suppose that we are "turning on" a drone that was powered down. What events might occur, and how would you create a state machine for this case? How would you use the function server to implement this state machine? Where will the state be stored?
Lecture 4	This lecture was really a recap reviewing what we learned in lectures 1-3 and walking through the decision-making process relative to a farming scenario. It didn't have separate content that should be learned, beyond what was listed above. [This isn't a question, just an observation.]
Lecture 5	What is the Azure IoT Hub? Why do cloud systems tend to block attempts to make a TCP connection from inside the cloud to something external to the cloud, in the Internet? Why can the Azure IoT Hub make connections if your code can't? What security properties can the Azure IoT Hub "guarantee"? Suppose that a software patch is required but the Azure IoT Hub has no connection to the sensor right now. How do they handle that? Explain the difference between two cases: sensors that have a known limit on their measurement quality, versus sensors that are faulty. Define temporal accuracy and precision, and give one example of each. Define clock skew. If you know the accuracy and precision for a set of sensors, what does this imply about the clock skew for those sensors? Suppose that we have 3 sensors measuring the same property, and we know that at most 1 of the 3 is faulty. How can we use this knowledge to exclude "extremely faulty" data? Suppose that 2 of 3 sensors overlap but 1 sensor doesn't overlap. Can we use this to improve the quality of our sensor measurement? What if all 3 sensors overlap but 1 might be faulty. In the worst case, how much can the faulty sensor "fool us" about the correct sensor reading? How can machine learning further improve the quality of a sensor estimate? Explain why the statement "If the temperature is above 70, turn on the A/C" is ambiguous. What possible interpretations arise? You can reference our 3 sensors.
Lecture 6	Why does Lamport believe that clock time cannot be used to reach conclusions about the causality of a set of events. Give an example. What is the English-language meaning of the $\rightarrow$ symbol, as defined by Lamport? What does it mean, in English, to say that events A and B are concurrent? Write a definition of "concurrent" for events A and B, using Lamport's $\rightarrow$ symbol. Suppose that a distributed system has processes {P, Q, …. } (N of them in total). How much space is required to implement logical timestamps (at each process, and also added to each message)? Now tell us how much space would be needed for vector timestamps? Suppose that a lot of events occur at P and its logical clock reads 189971. Then A occurs at P. Meanwhile, Q has been idle, and as a result, Q's logical clock reads 5. Now B occurs at Q. Draw a space-time diagram, labelling A and B with their logical times, for the case where no messages are sent between P and Q. Now modify the space-time diagram so that one message is sent from Q to P (yes, Q to P), at time 7, and received at time 10101 on P. Did this force you to change the logical time shown for A or for B? Explain why, or why not.

	Now reverse the direction: P sent the message to Q when P's clock was 10101, and Q's was 7 (so Q received it at time 7, before B occurs). Does this force any changes to the timestamps shown for A or B? Explain. Repeat the same sequence but with vector timestamps. Draw a space-time diagram for a system in which there are many possible "combinations" of events that all would have fall into the same real-time period because of clock skew. Your diagram should include some message passing events (sends and receives). Define "consistent cut" and "consistent snapshot" In your space-time diagram, use one color to draw some consistent cuts. Then use a second color to draw some cuts that are not consistent. Describe some properties that can be evaluated safely (correctly) on a consistent cut, but where a mistake can occur if the same test occurs on an inconsistent cut. Describe an IoT scenario in which you would want to use a consistent cut.
Lecture 7	What is the HDFS μ-service, and where is it normally used? HDFS includes a normal POSIX file system API, plus a snapshot operation. What does the snapshot operation do? Sally thinks that HDFS only supports file append operations. Sam thinks this is not true and that with the file system seek command, you can write code that would overwrite blocks in the middle of a file, not just append. Who is right? How does the answer to the question about file appends help us understand how the HDFS snapshot operation could be implemented? Consider the case from the questions about Lecture 6, where there are many events that all seem to have the same timestamp. Suppose those events are file system writes at time T, and now you do a snapshot in HDFS at time T. What would HDFS include in the snapshot? Suppose that sensors that have clocks are generating IoT records that are written into HDFS files. Same question: we do a snapshot at time T. Will the sensor time be used by HDFS to decide which file records to include in the snapshot? Freeze Frame File System (FFFS) is built as a modification of HDFS. List some of the ways it differs from HDFS. FFFS uses a mixture of clock timestamps and logical timestamps (we didn't discuss the details). Give one example of how the realtime (clock) value turns out to be relevant to the way FFFS behaves, and one example of how the logical clock value is used. Draw a spacetime diagram in which there are many possible consistent cuts, where time could "advance" and yet the consistent cuts might cross so that one cut includes event A and not B, and some other cut includes B and not A. Would this matter? FFFS guarantees that if T' > T, than the data returned by FFFS at time T' will include all events that would be included for time T. Why would this be useful? Give a practical illustration of these two points (the ones immediately above) in a farm drone scenario, where the IoT system would "work better" if it has the FFFS behavior and worse if it has the HDFS behavior. The lecture mentions Derecho's object store. We will see this again in Lecture 10.
Lecture 8	This lecture focused on the US air traffic control project. List some basic safety needs for an air traffic control system. How did the IBM architecture propose to implement a platform for air traffic control? How did the IBM architecture plan to track "variables" like airplane flight plans? How did the IBM architecture plan to handle periodic events? What was the model adopted by IBM?

	What does it mean to make assumptions about clocks, networks and failures, in the case of the CASD protocols? Is it reasonable to assume that "at most 3 crashes can occur in any 15s period", or "no more than 2 messages will be lost"? If it was your job to build an air traffic system, how could you come up with numbers for these kinds of assumptions? Would it help to be told to focus on a 99% confidence? Suppose a process is delayed by the operating system scheduler. For example, some message arrives from the network, but instead of waking up instantly to read it, the O/S delays it for 1s because the O/S was scheduling some other task, or paging, etc. No crash occurs, but this delay is longer than the assumption about how long it takes for 99% of all messages to get through. Is this a failure (but nothing crashed), or not? Give more examples of ways that non-faulty processes could accidentally violate the CASD assumptions. How would CASD "model" such events? Does CASD tell a process that it is correct, or that it is faulty? Explain. CASD is an atomic multicast protocol. What properties does it offer? Suppose that a process is faulty but not actually crashed. For example, it might have a malfunctioning clock. What properties does CASD guarantee for messages sent by such a process? What does it guarantee about messages delivered to such a process? Why does CASD have a "delta-T" delay built into the protocol? Why was the initial value of the CASD delta-T delay so large (30s – 3m)? How does the value of delta-T impact the end-user experience for an air traffic control system? Give an example involving a request by a pilot. In class we discussed the distinction between consistency first and realtime first. Which approach is being used in CASD? Explain why your answer is correct. Consider the requirement that "there should be exactly one controller for each plane." Is this a consistency or a realtime requirement? If we had a variable in memory that tells which controller "owns" some flight plan, could CASD be used to update the variable when the plane is handed off from one controller to a different one? Why, or why not?
Lecture 9	This lecture was another "big picture" recap. In some ways it was similar to lecture 4. The basic idea was similar: walk through more stories about IoT use cases on farms, and try to identify ways to break the resulting tasks into a set of subtasks, then map the subtasks to functions in the Azure IoT Hub function server or to μ-services (ones that exist or new ones you might need to build). Note: The main difference between this and lecture 4 was that we are familiar now with the architecture of modern IoT Cloud platforms like Azure, and as a result some of the confusing elements should be a bit more familiar. That lets us think more about the puzzle of moving machine intelligence from the back-end platforms to the edge, into new and specialized μ-services. But building those services is a bit of a puzzle because Azure lacks really sophisticated tools for helping you with basic aspects like bootstrap, repairing the state if the μ-service had stored data on files or in a database, managing membership of the μ-service (which processes are in it, and what their IP addresses are), Fault-tolerant state replication, etc. [This isn't a question, just an observation.] Not everything should move from the back-end to the edge. What are examples of things that should occur at the edge and probably need to move? What are examples of things that should remain in the back-end? At the end of the lecture there were some slides about what happens when we shift machine learning to the edge. List some requirements that arise.

	There is a slide late in the lecture about a pattern called MapReduce. The lecture talks about data being "always sharded". Do these two ideas have any connection? Why or why not? If Data is in a blob store or the Derecho object store, and we wanted to run MapReduce on the data, would it matter if the computational steps (the "map" and "reduce" logic) runs where the data is stored? Or could we write a normal program in a language like Python, pull the data over, and run the MapReduce logic wherever the program is running? Explain your answer carefully so that we can understand the major concerns that lead you to answer the way you do. Suppose that a single application seems to have needs for consistency, fault-tolerance realtime, automated restart after crashes, elasticity, and scalability through sharding. Why might some of these goals potentially be "at odds" with other goals? If you did have a need to create a μ-server with several kinds of properties all at once, how would it help if you could easily structure it into subsystems that each handles a different aspect? Why might you still prefer to call this a single μ-service and not just build a bunch of different μ-services that each treats the others as a black box – callable via remote procedure calls but with a completely hidden implementation.
Lecture 10	Concept of managed membership. Idea of linking to a library that helps your N processes form a group with N members that can contact one-another. Derecho is such a library. Concept of state machine replication. Example: chain replication (useless without a membership solution). List some properties required from a membership service. Definition of "split brain" behavior caused by a network partitioning event (link failure). Show that requiring a quorum permits progress without allowing a split brain scenario. Give an example in which processes didn't crash, but this same rule blocks progress. Properties of a "totally ordered multicast" protocol: even if two messages are sent at the same time, they are delivered in order. How to use priority queues and logical clocks to create a totally ordered multicast protocol. Paxos properties (ordering, durability, exactly-once delivery), but no details. In fact the slide deck does include details (we didn't cover those slides in class), for people who are curious, but it is not required material. Derecho uses "round robin" ordering. It can do this because the epoch has a well-known membership view, and it includes a list of which processes are senders in the view. Understanding space-time diagraphs for groups with join, state transfer, multicast, crash. Lamport's Paxos "model". We won't look at his actual protocol. Beyond total order, it deals with failures and also with saving data on disk and recovering it after a crash. Derecho's use of virtual synchrony epochs. Concept of RDMA (remote direct memory access) hardware. Derecho's way of moving data on RDMA. Derecho's failure handling approach is to clean up disrupted multicasts, in a way that fits the Paxos model. Performance of Derecho on RDMA is very impressive, but it also works on normal TCP. The Derecho object store is a library within Derecho built on the Derecho multicast and point-to-point messaging layer. A library built on a library! You can use it as a library, or can set it up to run as a free-standing μ-service.
Lecture 11	Georeplication

	Availability zones NAT IP translation, TCP connectivity limitations Zone Aware services Spanner, true time, consistency from order-based WAN mirroring 5G is very much like IoT Multipath TCP Thought question: We learned about how Spanner is using TT to order transactions. But suppose we wanted an additional guarantee, namely that if at *any* site transaction A $\rightarrow$ B (meaning that all the work A did happened before any of the work B will do) then Spanner will provide this property at *every* site. Notice that this is a statement about causal order and consistency, not real-time, but that TT is a mechanism that provides a real-time guarantee. Would Spanner's current TT-based implementation provide this guarantee? If not, give a counter-example. If so, explain why. Another thought question: Suppose that in availability zone Za there is a zone-aware service that carries messages to zone Zb for delivery. The service uses N side-by-side TCP connections and it basically spreads the outgoing messages evenly so that every TCP connection has a similar level of load, and each is carrying distinct messages. To preserver ordering when one sender sends a sequence of messages, the service also guarantees that if a single sender sent m1 and then m2, then m1 and m2 will be put on the same TCP connection, so that m2 will be sent after m1 and hence will arrive after m2. Now assume that we read about how Spanner is using TT, and want to implement a mechanism similar to the one used by Spanner. Could an application use the Spanner TT idea in conjunction with this zone-aware mechanism? In thinking about this, assume that your application will be told which connection each incoming message showed up on, so that if there are N connections from A to B, you'll know N and for each message will know which connection it showed up on.
Lecture 12	Concepts: Gossip protocol, fixed message rate, unreliable messages, push gossip, pull gossip, messages with a fixed maximum size. In class we noted that gossip can never exceed a known "configured" peak data rate and message rate. But we also noted that due to the birthday paradox, a process actually can receive more than one incoming gossip message per round. Why wouldn't this effect mean that the data rate and message rate can be higher than the configured rate? For example, if a process might receive 3 messages in one round, wouldn't this imply that the system can have a message rate 3x higher than the normal average? Gossip has a classic S-shaped infection curve. But suppose that in some system 50% of the messages are dropped by the network. How would the curve change? What about for 75% loss rate, or 90%? What mechanism does Bimodal Multicast combine with gossip, and why does this cause a delivery delay distribution with two modes (two "peaks")? Suppose that in Bimodal Multicast some region drops all copies of some message, so that only the original sender (call it process S) has a copy. What will the protocol do, and how might the delay distribution look relative to the typical one? In Astrolabe, a bound is imposed on the "region" size, and this forces the use of a tree structure (a hierarchical set of regions). Why not just have one region covering the entire data center? You can assume that a data center has at most 1M compute nodes. In thinking about this, consider (1) robustness, (2) speed of information spread, (3) average workload for each computer using Astrolabe.

Lecture 13	A BlockChain is really an append-only file protected by cryptographic signatures. Draw a picture of a BlockChain and in show where the cryptographic signatures would be stored in the file. Do this for the permissioned case, and then again for a BlockChain using the permissionless model. What is the difference between a permissioned and a non-permissioned BlockChain? Suppose that an operator of a permissioned BlockChain wanted to the modify record number 57, and was prepared to recompute all the signatures. As a customer of the company operating the BlockChain, what information should you store to be able to detect this? Now think about the same scenario with a permissionless BlockChain. In what ways is a permissionless BlockChain harder to modify than a permissioned one? What is a Merkle tree, and what is the purpose for organizing records into a Merkle tree? In several of the BlockChain standards records don't just record transfers of bitcoins. The fancy standards allow arbitrary code in a special language. Beyond the variables actually in the functions stored in these code blocks, what other data sources can they touch? If there is a restriction or limitation, why is that limitation imposed?
Lecture 14	List some risks associated with the Blockchain model. Be able to <u>define</u> each threat. We heard a story about how client-server computing stumbled because of unanticipated technical needs that early versions lacked. How do the BlockChain risks stack up: how many are total mysteries, and how many look like things that might just take time to do? If we wanted to log data for a sensor on a farm, what additional data would we need to log to be sure that the audit can show that this is the correct sensor for its role, is still in use, is accurate (time and value), is configured properly, has all need software patches and so on? Do you think this is really feasible? Suppose that you wanted absolute certainty that a block will not be rolled back. How long would you need to wait in a permissioned model? What about a permissionless model? Why does Vegvisir assume that network connectivity is intermittent? How does it solve this issue? Why does it favor "conflict-free" operations for what gets logged? Explain why the proof of work "concept" is really only relevant to wide-area Blockchain in the permissionless model. Suppose we wanted to keep an audit trail for events involving production of cheese or yoghurt on a farm and then in the unit that actually ferments the milk to make the cheese/yoghurt. Compared to a hand-written ledger, what advantages and special challenges would BlockChain introduce? Suppose that a Vegvisir chain records a photo from a farm and that someone tells you that the photo was taken by such and such a camera on such and such a day, located in a particular place at that time, etc. Now suppose that we want a "reasonable level of auditability" for these claims. What additional information would need to be logged?
Lecture 15	Hardware accelerators. Benefits: primarily cost per operation, not necessarily speed. GPU and TPU both are valuable for massively parallel operations, like on every pixel of a " photo or video. A general purpose CPU would need a loop that does things one by one. These devices have "single instruction, multiple data" capabilities. Many machine learning tasks include steps that are easy to express as parallel logic. By having the GPU or TPU perform them, we get a big speedup even though the GPU cycle time is not much better than the CPU (often, much worse, actually). TPU is like GPU but stripped down to essentials with just the bare bones for tensor math.

	Peak speed on GPU often requires coding in CUDA. For TPU, Tensor Flow. FPGA is a technology for creating a specialized chip on demand. It could be an extra CPU or a mini-GPU or TPU, or anything else you can design a circuit to do. FPGA is a large and slow kind of chip because it needs a way to dynamically install both the wiring diagram and the gate-level functions your code selects (your code would be written in Verilog). There is some recent work on compiling from higher level languages into FPGA, and at Cornell, Adrian Samson is an expert on that topic (his language is called Seashell). But you still need to think about code in a very different way. There is also work on operating systems to manage the FPGA itself. You can easily cause them to just hang and they would then need a power-down/reset/power-up. So we can't run code on them without extensive prior testing. Anything an FPGA can do could also be burned as an ASIC (an actual chip). This ASIC will be less flexible, but probably faster, smaller, and less power-hungry. We use FPGA for flexibility and to design ASICs, then eventually make ASICs if we will always need this specific logic and decide to put it right into permanent silicon. NVM means "non-volatile memory". A form of random access memory that won't lose what you store in it even if power goes down. New phase-change memory like 3-D Xpoint (Optane) implement a new form of NVM. It is direct byte-addressable memory with persistence. But there are other ways to build similar solutions. For example, we can memory-map blocks of a file and then that file becomes byte addressable from memory, and persistent. How would these newer solutions be different from that standard option? Why are data center owners so careful about who can use these hot new technologies? Why might this mean that only vendor-operated specialized services will access them?
Lecture 16	Long path to RDMA. RDMA emerged on Infiniband, like Ethernet but sender always knows that the receiver has preallocated space for the incoming object ("holds send credits") and hence no loss will occur due to a lack of space at the next hop. Adds up to a hop-by-hop pre-allocation of buffers from sender to the receiver, hence if the wire loses no packets, the transfer can be done totally reliably. And optical networks do not corrupt data (loss rate is like backplane bus failures: 10^-19 to 10^-21 probability per bit). HPC systems simply run Infiniband and Ethernet side by side and the RDMA runs on Infiniband. Doubles cost But RDMA on Ethernet (RoCE) has had more challenges. Idea is to run them side by side in a logical way, but this was hard. Priority pause frames (PPFs) can be generated at really high rates, like billions/second. DCQCN and TIMELY add end-to-end congestion windowing and this helps a lot. Microsoft decided to deploy RDMA on RoCE with DCQCN but it was very hard to pull off. Today they have RDMA in all data centers but only allow their own products to touch it. Someday maybe Derecho can do so too, once they gain confidence. For Azure HPC they still use a dual network with Infiniband. High-payoff technologies still can have a slow path to being widely adopted.
Lecture 17	Core idea: could we leverage a cloud for computation on private data like remarks made in a private space, or images? With an uncooperative operator there are many obstacles and probably it cannot be done even using Intel SGX, a hardware feature for precisely this case. SGX creates firewalled compute contexts that the owner of the computer can't peak into. You can run your computation there, and combine private data with big data sets from the cloud, for

	example, and can send back the results on an encrypted network connection. But there might still be risks of leakage through microservices you depend upon or ML models created as a side-effect of your computation. So it is probably impossible to get privacy from an untrustworthy vendor. What if a vendor like Microsoft were to promise "as much privacy as we can jointly offer" with a two-sided obligation: you as the user must do X, Y and Z and Microsoft for its side will do A, B and C? In this model (leave no trace behind) we could do far better. But sharing with untrusted third parties would remain a risk (ORAM could help but just not having to share would be far better). Micro-services will also need to be privacy preserving. Work at MIT showed an example of a privacy-preserving database layered on an unmodified standard database, called CryptDB. We reviewed the slides. Key ideas: onion encryption, multiple cryptographic methods used for different onion layers, some methods turn out to still allow certain mathematical operations even on encrypted data, like "less then" or "equality". CryptDB packages the resulting solution and is an open source, open developer tool you can use.
Lecture 18	Prelim exam. There was no lecture on this date.
Lecture 19. **This would not be seen on an exam in cs5412, but you may be asked about these ideas during project demos**	Why is big IoT data not identical to classic big data? Always sharded, all the time Refresher on main ideas from machine learning, analogy to spline fitting. Facebook TAO: role is to track the continuously evolving social network graph. Why is TAO not just a single database with transactions against it? Model: an official backend database with edge infrastructure that updates continuously Flow of data in TAO, handling of failures. Long latencies. Concept of a provisional update done locally and an official update that might revise the graph and occurs later. Convergent consistency. Will IoT systems be "more like TAO" or "more like standard web style big data"? How will the evolution of the market shape opportunities to earn revenue, and by doing so, point us to the likely early technology opportunities? Who will invest "first"?