CS162
Operating Systems and Systems Programming
Lecture 25

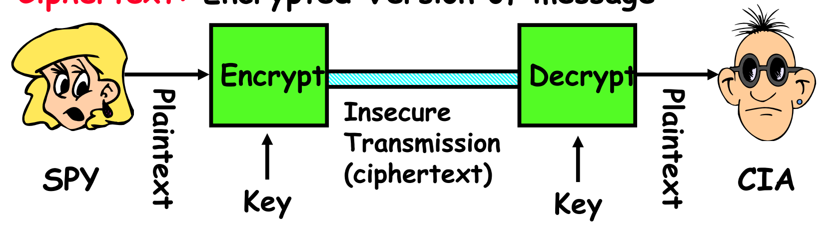
A Touch more Security and Quantum Computing + IoT

December 7th, 2015

Prof. John Kubiatowicz
http://cs162.eecs.Berkeley.edu

Recall: Private Key Cryptography

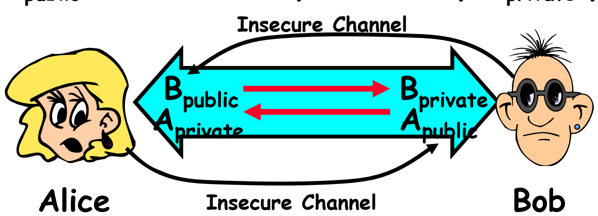
- Private Key (Symmetric) Encryption:
  - Single key used for both encryption and decryption
- Plaintext: Unencrypted Version of message
- Ciphertext: Encrypted Version of message



- Important properties
  - Can't derive plain text from ciphertext (decode) without access to key
  - Can't derive key from plain text and ciphertext
  - As long as password stays secret, get both secrecy and authentication
- Symmetric Key Algorithms: DES, Triple-DES, AES

Recall: Public Key Encryption Details

- Idea: K_{public} can be made public, keep $K_{private}$ private

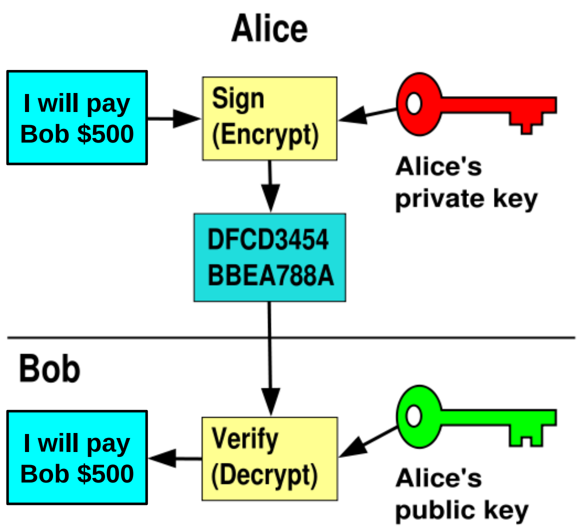


- Gives message privacy (restricted receiver):
  - Public keys (secure destination points) can be acquired by anyone/used by anyone
  - Only person with private key can decrypt message
- What about authentication?
  - Use combination of private and public key
  - Alice→Bob: [(I'm Alice)^{Aprivate} Rest of message]^{Bpublic}
  - Provides restricted sender and receiver
- But: how does Alice know that it was Bob who sent her B_{public}? And vice versa…
  - Need a certificate authority to sign keys!

Non-Repudiation: RSA Crypto & Signatures

- Suppose Alice has published public key KE
- If she wishes to prove who she is, she can send a message x encrypted with her private key KD (i.e., she sends E(x, KD))
  - Anyone knowing Alice's public key KE can recover x, verify that Alice must have sent the message
    - » It provides a signature
  - Alice can't deny it ⇒ non-repudiation
- Could simply encrypt a hash of the data to sign a document that you wanted to be in clear text
- Note that either of these signature techniques work perfectly well with any data (not just messages)
  - Could sign every datum in a database, for instance

RSA Crypto & Signatures (cont'd)



Digital Certificate Authorities

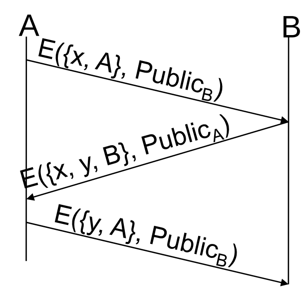
- How do you know K_E is Alice's public key?
- Trusted authority (e.g., Verisign) signs binding between Alice and K_E with its private key $KV_{private}$
  - C = E({Alice, K_E}, $KV_{private}$)
  - C: digital certificate
- Alice: distribute her digital certificate, C
- Anyone: use trusted authority's KV_{public}, to extract Alice's public key from C
  - D(C, KV_{public}) = D(E({Alice, K_E}, $KV_{private}$), KV_{public}) = {Alice, K_E}
- Where does someone get KV_{public} from?
  - Typically compiled into the browser (for instance)!
  - Can you trust this??

Properties of RSA Public Cryptosystems

- Requires generating large, random prime numbers
  - Algorithms exist for quickly finding these (probabilistic!)
- Requires exponentiating very large numbers
  - Again, fairly fast algorithms exist
- Overall, much slower than symmetric key crypto
  - One general strategy: use public key crypto to exchange a (short) symmetric session key
    - » Use that key then with AES or such
- How difficult is recovering d, the private key?
  - Equivalent to finding prime factors of a large number
    - » Many have tried - believed to be very hard (= brute force only)
    - » (Though quantum computers could do so in polynomial time!)

Simple Public Key Authentication

- Each side need only to know the other side's public key
  - No secret key need be shared
- A encrypts a nonce (random num.) x
  - Avoid replay attacks, e.g., attacker impersonating client or server
- B proves it can recover x, generates second nonce y
- A can authenticate itself to B in the same way
- A and B have shared private secrets on which to build private key!
  - We just did secure key distribution!
- *Many more details to make this work securely in practice!*



Notation: E(m,k) – encrypt message m with key k

Summary of Our Crypto Toolkit

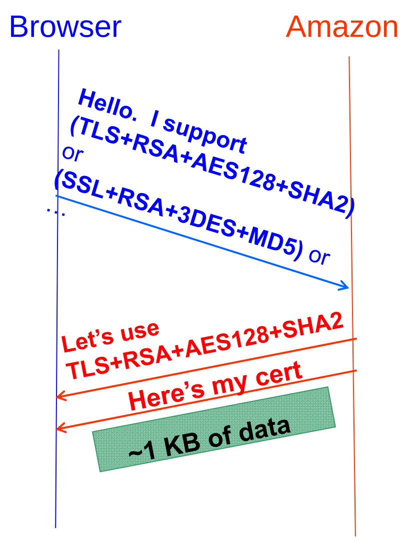
- If we can securely distribute a key, then
  - Symmetric ciphers (e.g., AES) offer fast, presumably strong confidentiality
- Public key cryptography does away with (potentially major) problem of secure key distribution
  - But: not as computationally efficient
    - » Often addressed by using public key crypto to exchange a session key
- Digital signature binds the public key to an entity
- Public Key Pairs can serve as Identities!
  - Verified by certificate authority
  - Or distributed by other techniques

Putting It All Together - HTTPS

- What happens when you click on https://www.amazon.com?
- `https` = "Use HTTP over SSL/TLS"
  - SSL = Secure Socket Layer
  - TLS = Transport Layer Security
    - » Successor to SSL
  - Provides security layer (authentication, encryption) on top of TCP
    - » Fairly transparent to applications

HTTPS Connection (SSL/TLS) (cont'd)

- Browser (client) connects via TCP to Amazon's HTTPS server
- Client sends over list of crypto protocols it supports
- Server picks protocols to use for this session
- Server sends over its certificate
- (all of this is in the clear)

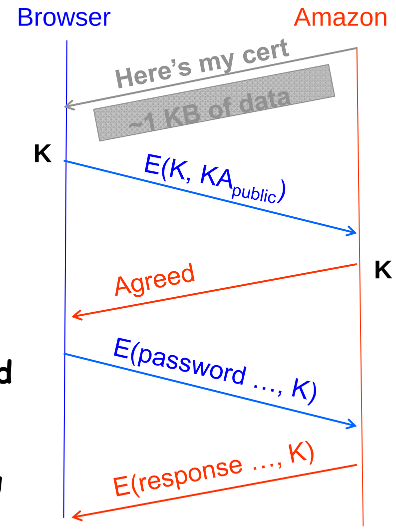


Inside the Server's Certificate

- Name associated with cert (e.g., Amazon)
- Amazon's RSA public key
- A bunch of auxiliary info (physical address, type of cert, expiration time)
- Name of certificate's signatory (who signed it)
- A public-key signature of a hash (SHA-256) of all this
  - Constructed using the signatory's private RSA key, i.e.,
  - Cert = $E(H_{SHA256}(KA_{public}$, www.amazon.com, …), $KS_{private}))$
    - » KA_{public}: Amazon's public key
    - » $KS_{private}$: signatory (certificate authority) private key
- …

HTTPS Connection (SSL/TLS) cont'd

- Browser constructs a random *session key* K used for data communication
  - Private key for bulk crypto
- Browser encrypts K using Amazon's public key
- Browser sends E(K, KA_{public}) to server
- Browser displays 
- All subsequent comm. encrypted w/ symmetric cipher (e.g., AES128) using key K
  - E.g., client can authenticate using a password



Administrivia

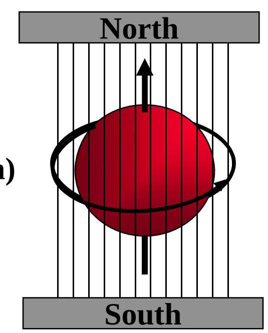
- Midterm 2 grading still continuing
  - ETA: very soon.
  - Have a couple of sub problems still to grade
  - Solutions have been posted
- Final Exam
  - Friday, December 18th, 2015.
  - 3-6P, Wheeler Auditorium
  - All material from the course
    - » (excluding option lecture on 12/7)
    - » With slightly more focus on second half, but you are still responsible for all the material
  - Two sheets of notes, both sides
  - Will need dumb calculator
- Targeted review sessions: See posts on Piazza
  - Possibly 3 different sessions focused on parts of course

Use Quantum Mechanics to Compute?

- Weird but useful properties of quantum mechanics:
  - Quantization: Only certain values or orbits are good
    - » Remember orbitals from chemistry???
  - Superposition: Schizophrenic physical elements don't quite know whether they are one thing or another
- All existing digital abstractions try to eliminate QM
  - Transistors/Gates designed with classical behavior
  - Binary abstraction: a "1" is a "1" and a "0" is a "0"
- Quantum Computing:
  Use of Quantization and Superposition to compute.
- Interesting results:
  - Shor's algorithm: factors in polynomial time!
  - Grover's algorithm: Finds items in unsorted database in time proportional to square-root of n.
  - Materials simulation: exponential classically, linear-time QM

Quantization: Use of "Spin"

Spin ½ particle:
(Proton/Electron)



Representation:
|0> or |1>

- Particles like Protons have an intrinsic "Spin" when defined with respect to an external magnetic field
- Quantum effect gives "1" and "0":
  - Either spin is "UP" or "DOWN" nothing between

Now add Superposition!

- The bit can be in a combination of "1" and "0":
  - Written as: $\Psi = C_0|0\rangle + C_1|1\rangle$
  - The C's are *complex numbers!*
  - Important Constraint: $|C_0|^2 + |C_1|^2 = 1$
- If *measure* bit to see what looks like,
  - With probability $|C_0|^2$ we will find $|0\rangle$ (say "UP")
  - With probability $|C_1|^2$ we will find $|1\rangle$ (say "DOWN")
- Is this a real effect? Options:
  - This is just statistical – given a large number of protons, a fraction of them ($|C_0|^2$) are "UP" and the rest are down.
  - This is a real effect, and the proton is really both things until you try to look at it
- Reality: second choice!
  - There are experiments to prove it!

A register can have many values!

- Implications of superposition:
  - An *n*-bit register can have 2^n values simultaneously!
  - 3-bit example:

$$\Psi = C_{000}|000\rangle + C_{001}|001\rangle + C_{010}|010\rangle + C_{011}|011\rangle + C_{100}|100\rangle + C_{101}|101\rangle + C_{110}|110\rangle + C_{111}|111\rangle$$

- Probabilities of measuring all bits are set by coefficients:
  - So, prob of getting $|000\rangle$ is $|C_{000}|^2$, etc.
  - Suppose we measure only one bit (first):
    - » We get "0" with probability: $P_0=|C_{000}|^2+|C_{001}|^2+|C_{010}|^2+|C_{011}|^2$
      Result: $\Psi = (C_{000}|000\rangle + C_{001}|001\rangle + C_{010}|010\rangle + C_{011}|011\rangle)$
    - » We get "1" with probability: $P_1=|C_{100}|^2+|C_{101}|^2+|C_{110}|^2+|C_{111}|^2$
      Result: $\Psi = (C_{100}|100\rangle + C_{101}|101\rangle + C_{110}|110\rangle + C_{111}|111\rangle)$
- Problem: Don't want environment to *measure* before ready!
  - Solution: Quantum Error Correction Codes!

Spooky action at a distance

- Consider the following simple 2-bit state:

  $\Psi = C_{00}|00\rangle + C_{11}|11\rangle$
  - Called an "EPR" pair for "Einstein, Podolsky, Rosen"
- Now, separate the two bits:

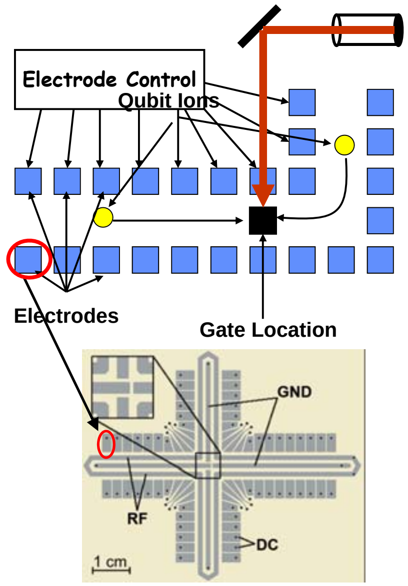
- If we measure one of them, it instantaneously sets other one!
  - Einstein called this a "spooky action at a distance"
  - In particular, if we measure a $|0\rangle$ at one side, we get a $|0\rangle$ at the other (and vice versa)
- Teleportation
  - Can "pre-transport" an EPR pair (say bits X and Y)
  - Later to transport bit A from one side to the other we:
    - » Perform operation between A and X, yielding two classical bits
    - » Send the two bits to the other side
    - » Use the two bits to operate on Y
    - » Poof! State of bit A appears in place of Y

MEMs-Based Ion Trap Devices

- Ion Traps: One of the more promising quantum computer implementation technologies
  - Built on Silicon
    - » Can bootstrap the vast infrastructure that currently exists in the microchip industry
  - Seems to be on a "Moore's Law" like scaling curve
    - » 12 bits exist, 30 promised soon, …
    - » Many researchers working on this problem
  - Some optimistic researchers speculate about room temperature
- Properties:
  - Has a long-distance Wire
    - » So-called "ballistic movement"
  - Seems to have relatively long decoherence times
  - Seems to have relatively low error rates for:
    - » Memory, Gates, Movement

Quantum Computing with Ion Traps

- Qubits are atomic ions (e.g. Be^{+})
  - State is stored in hyperfine levels
  - Ions suspended in channels between electrodes
- Quantum gates performed by lasers (either one or two bit ops)
  - Only at certain trap locations
  - Ions move between laser sites to perform gates
- Classical control
  - Gate (laser) ops
  - Movement (electrode) ops
    - Complex pulse sequences to cause Ions to migrate
    - Care must be taken to avoid disturbing state
- Demonstrations in the Lab
  - NIST, MIT, Michigan, many others



Courtesy of Chuang group, MIT

Model: Operations on coefficients + measurements

Input Complex State → Unitary Transformations → Measure → Output Classical Answer

- Basic Computing Paradigm:
  - Input is a register with superposition of many values
    - » Possibly all 2n inputs equally probable!
  - Unitary transformations compute on coefficients
    - » Must maintain probability property (sum of squares = 1)
    - » Looks like doing computation on all 2n inputs simultaneously!
  - Output is one result attained by measurement
- If do this poorly, just like probabilistic computation:
  - If 2n inputs equally probable, may be 2n outputs equally probable.
  - After measure, like picked random input to classical function!
  - All interesting results have some form of "fourier transform" computation being done in unitary transformation

Shor's Factoring Algorithm

- The Security of RSA Public-key cryptosystems depends on the difficulty of factoring a number N=pq (product of two primes)
  - Classical computer: sub-exponential time factoring
  - Quantum computer: polynomial time factoring
- Shor's Factoring Algorithm (for a quantum computer)

Easy 1) Choose random x : $2 \le x \le N-1$.
Easy 2) If $\gcd(x,N) \neq 1$, Bingo!
Hard 3) Find smallest integer r : $x^r \equiv 1 \pmod N$
Easy 4) If r is odd, GOTO 1
Easy 5) If r is even, $a \equiv x^{r/2} \pmod N \Rightarrow (a-1)\times(a+1) = kN$
Easy 6) If $a \equiv N-1 \pmod N$ GOTO 1
Easy 7) ELSE $\gcd(a \pm 1, N)$ is a non trivial factor of N.

Finding r with $x^r \equiv 1 \pmod N$

$$\sum_k |k\rangle|1\rangle \longrightarrow \sum_k |k\rangle|x^k\rangle$$

$$= \sum_{w=0}^{r-1}\sum_y |w + r\,y\rangle|x^w\rangle$$

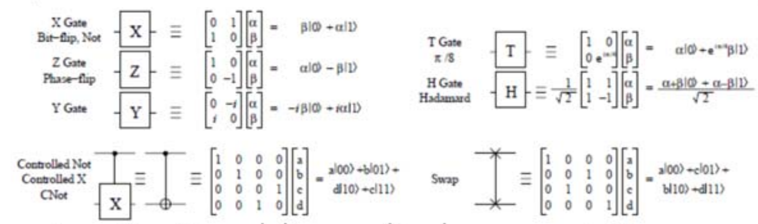
$$\xrightarrow{\text{Quantum Fourier Transform}} \sum_{w=0}^{r-1}\Big(\;\; \tfrac{0}{r} \;\; \tfrac{1}{r} \;\; \cdots \;\; \tfrac{k}{r} \;\; \Big)|x^w\rangle$$

- Finally: Perform measurement
  - Find out r with high probability
  - Get $|y\rangle|a^{w'}\rangle$ where y is of form k/r and w' is related

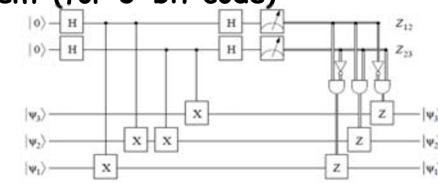
Quantum Computing Architectures

- Why study quantum computing?
  - Interesting, says something about physics
    - » Failure to build ⇒ quantum mechanics wrong?
  - Mathematical Exercise (perfectly good reason)
  - Hope that it will be practical someday:
    - » Shor's factoring, Grover's search, Design of Materials
    - » Quantum Co-processor included in your Laptop?
- To be practical, will need to hand quantum computer design off to classical designers
  - Baring Adiabatic algorithms, will probably need 100s to 1000s (millions?) of working logical Qubits ⇒ 1000s to millions of physical Qubits working together
  - Current chips: ~1 billion transistors!
- Large number of components is realm of *architecture*
  - What are optimized structures of quantum algorithms when they are mapped to a physical substrate?
  - Optimization not possible by hand
    - » Abstraction of elements to design larger circuits
    - » Lessons of last 30 years of VLSI design: USE CAD

Quantum Circuit Model

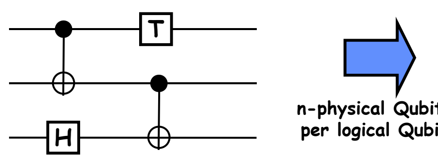


- Quantum Circuit model – graphical representation
  - Time Flows from left to right
  - Single Wires: persistent Qubits, Double Wires: classical bits
    - » Qubit – coherent combination of 0 and 1: $\psi = \alpha|0\rangle + \beta|1\rangle$
  - Universal gate set: Sufficient to form all unitary transformations
- Example: Syndrome Measurement (for 3-bit code)
  - Measurement (meter symbol) produces classical bits

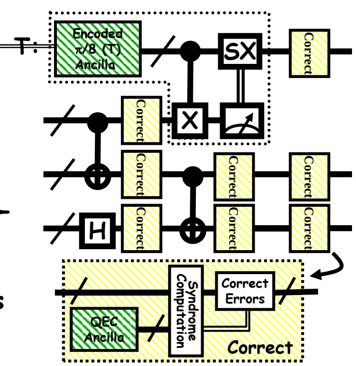


- Quantum CAD
  - Circuit expressed as netlist
  - Computer manpulated circuits and implementations

Adding Quantum ECC



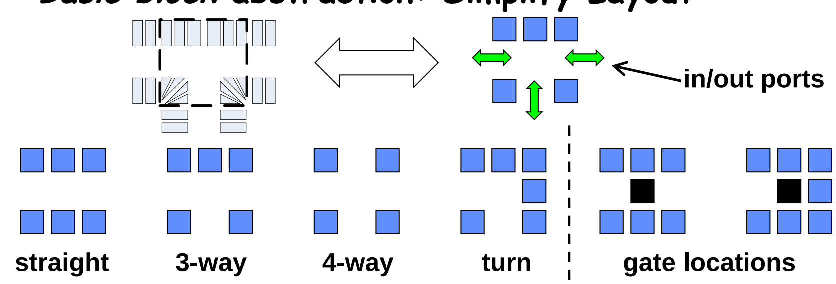
n-physical Qubits per logical Qubit



- Quantum State Fragile ⇒ encode all Qubits
  - Uses many resources: e.g. 3-level [[7,1,3]] code 343 physical Qubits/logical Qubit)!
- Still need to handle operations (fault-tolerantly)
  - *Some* set of gates are simply "transversal:"
    - » Perform identical gate between each physical bit of logical encoding
  - Others (like T gate for [[7,1,3]] code) cannot be handled transversally
    - » Can be performed fault-tolerantly by preparing appropriate ancilla
- Finally, need to perform periodical error correction
  - Correct after every(?): Gate, Long distance movement, Long Idle Period
  - Correction reducing entropy ⇒ Consumes Ancilla bits
- Observation: ≥ 90% of QEC gates are used for ancilla production
  ≥ 70-85% of all gates are used for ancilla production

An Abstraction of Ion Traps

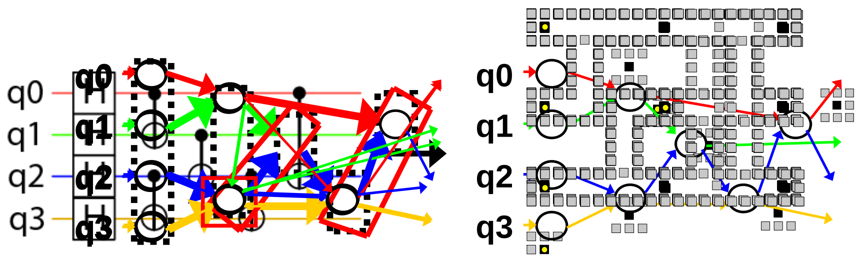
- *Basic block* abstraction: Simplify Layout



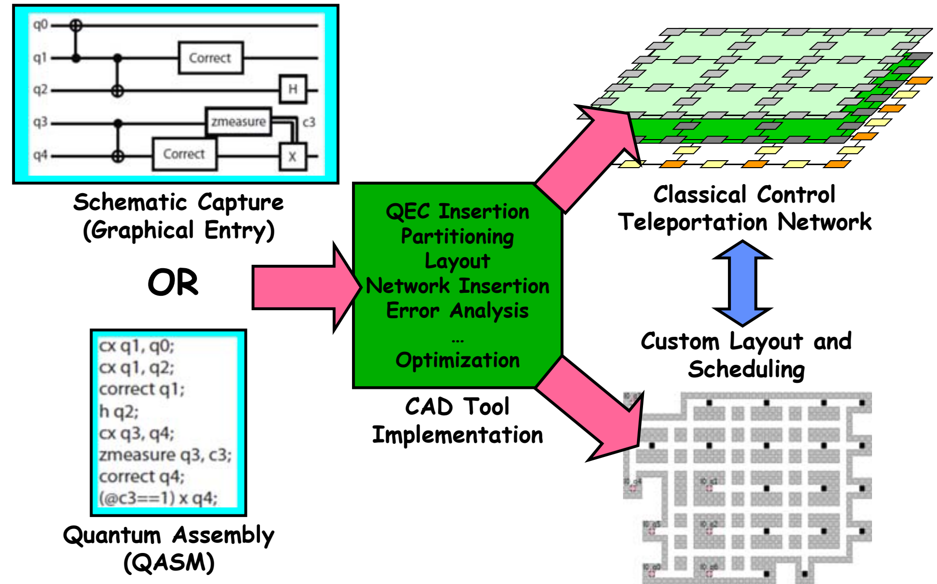
- Evaluation of layout through simulation
  - Yields Computation Time and Probability of Success
- Simple Error Model: Depolarizing Errors
  - Errors for every Gate Operation and Unit of Waiting
  - Ballistic Movement Error: Two error Models
    1. Every Hop/Turn has probability of error
    2. Only Accelerations cause error

Example Place and Route Heuristic: Collapsed Dataflow

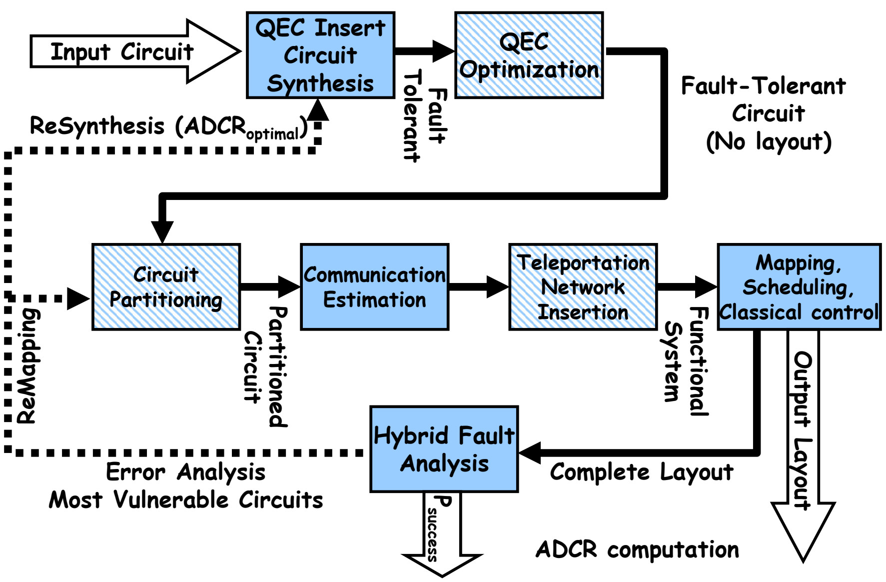
- Gate locations placed in dataflow order
  - Qubits flow left to right
  - Initial dataflow geometry folded and sorted
  - Channels routed to reflect dataflow edges
- Too many gate locations, collapse dataflow
  - Using scheduler feedback, identify latency critical edges
  - Merge critical node pairs
  - Reroute channels
- Dataflow mapping allows pipelining of computation!



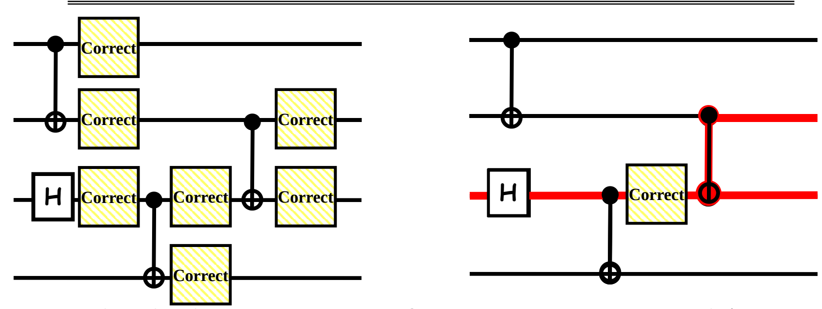
Vision of Quantum Circuit Design



Quantum CAD flow

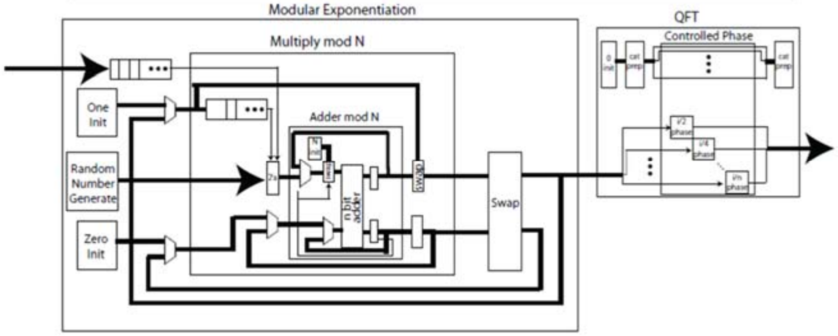


Sample Optimization: Reducing QEC Overhead



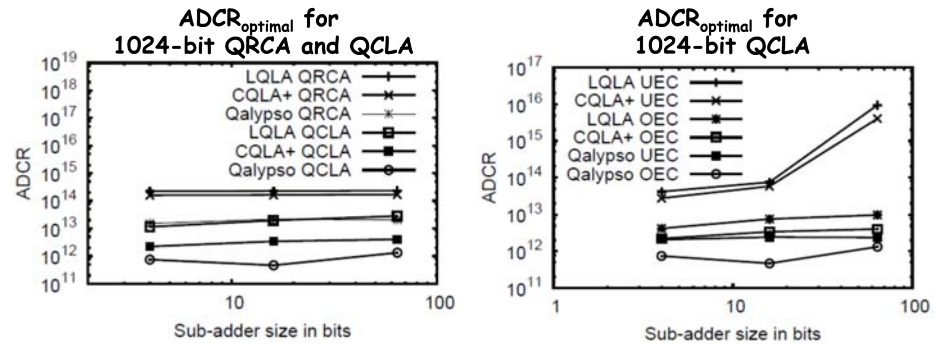
- Standard idea: correct after every gate, and long communication, and long idle time
  - This is the easiest for people to analyze
- This technique is suboptimal (at least in some domains)
  - Not every bit has same noise level!
- Different idea: identify critical Qubits
  - Try to identify paths that feed into noisiest output bits
  - Place correction along these paths to reduce maximum noise

Investigating 1024-bit Shor's



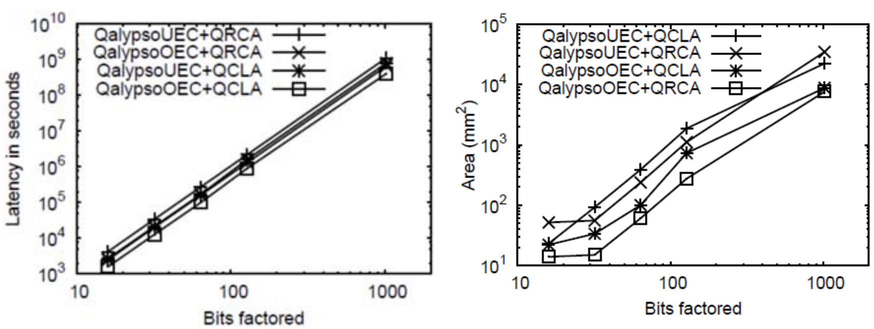
- Full Layout of all Elements
  - Use of 1024-bit Quantum Adders
  - Optimized error correction
  - Ancilla optimization and Custom Network Layout
- Statistics:
  - Unoptimized version: 1.35×10^{15} operations
  - Optimized Version 1000X smaller
  - QFT is only 1% of total execution time

Comparison of 1024-bit adders



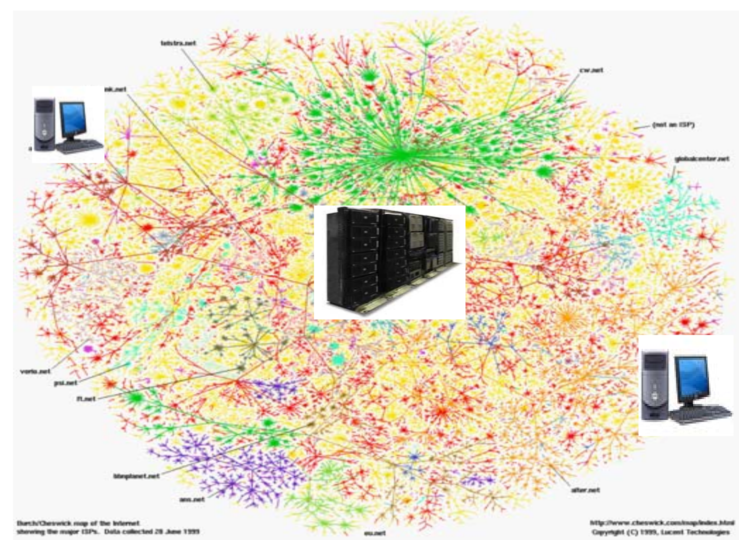
- 1024-bit Quantum Adder Architectures
  - Ripple-Carry (QRCA)
  - Carry-Lookahead (QCLA)
- Carry-Lookahead is better in all architectures
- QEC Optimization improves ADCR by order of magnitude in some circuit configurations

1024-bit Shor's Continued

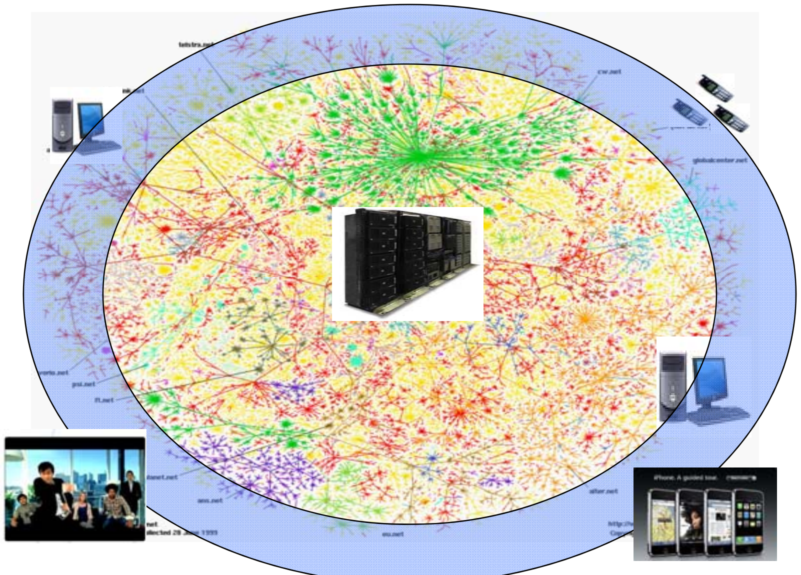


- Circuits too big to compute $P_{success}$
  - Working on this problem
- Fastest Circuit: 6×10^8 seconds ~ 19 years
- Smallest Circuit: 7659 mm^2
  - Compare to previous *estimate* of 0.9 m^2 = 9×10^5 mm^2

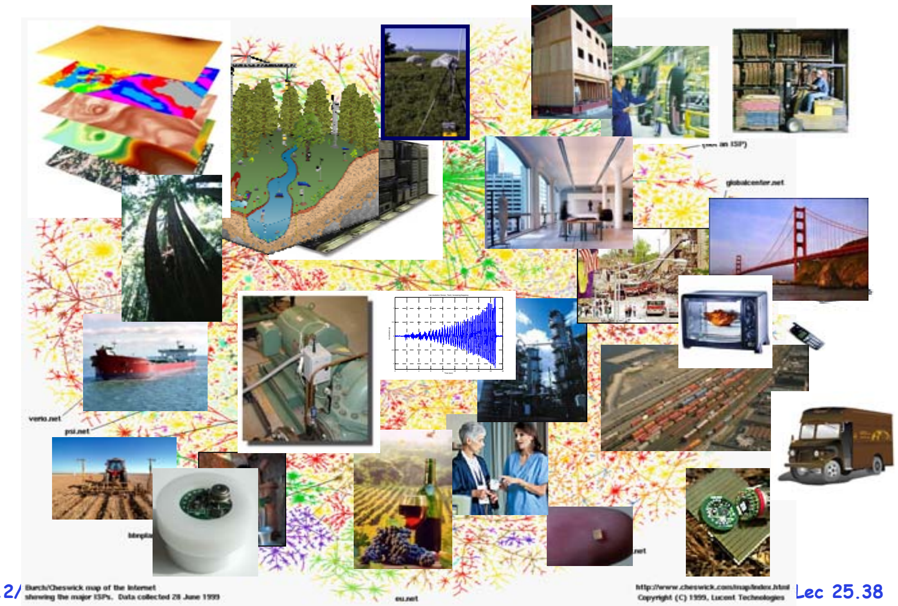
1997 - The Internet of Every Computer



2007 - The Internet of Every Body



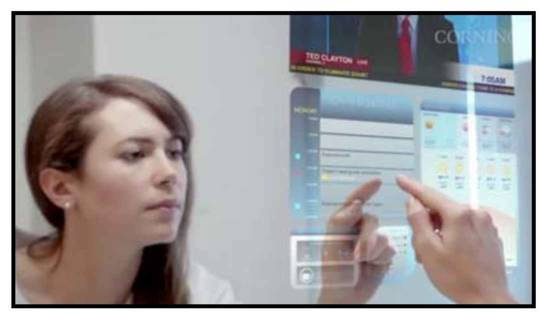
2017 - The Internet of Everyday Things



Why "Real" Information is so Important

Resources in a Smart Space (2011 Corning Glass)

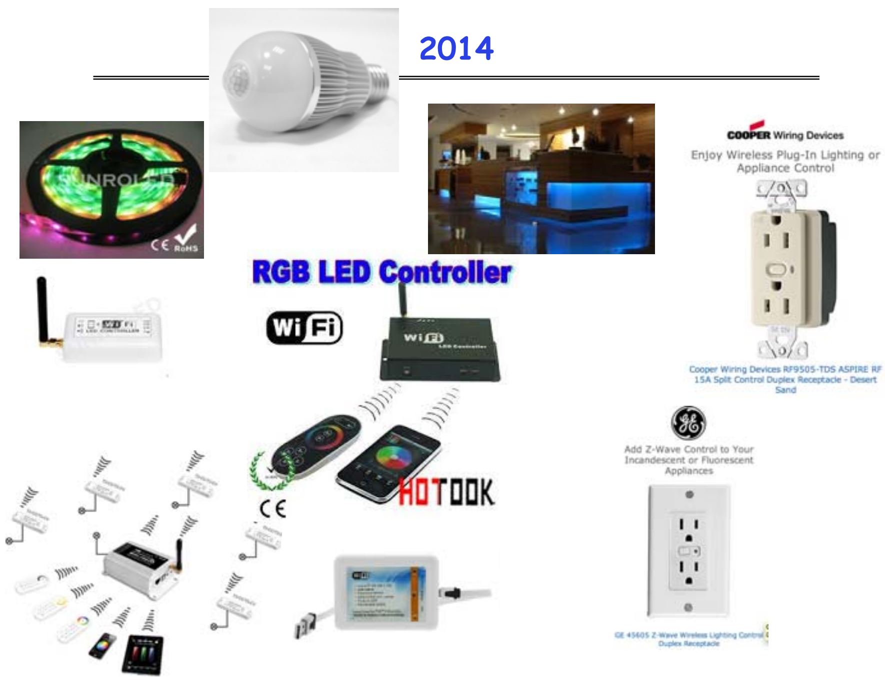
- Potential Displays Everywhere
  - Walls, Tables, Appliances, Smart Phones, Google Glasses….
- Audio Output Everywhere
- Inputs Everywhere
  - Touch Surfaces
  - Cameras/ Gesture Tracking
  - Voice
- Context Tracking
  - Who is Where
  - What do they want
  - Which Inputs map to which applications



A Day Made of Glass ©Corning

2013

2014



2013

The Nest makes headlines!

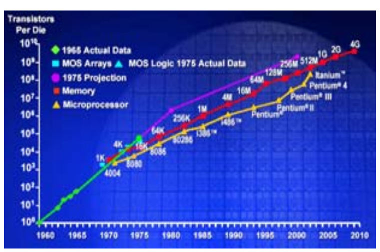
2014

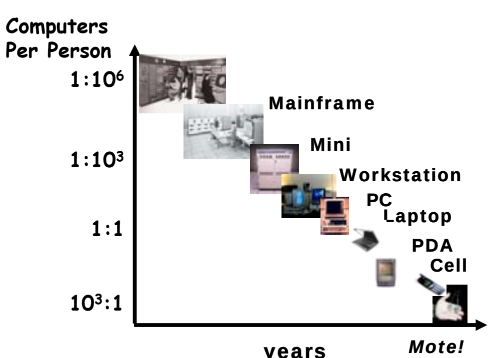
2014

Broad Technology Trends

Moore's Law: # transistors on cost-effective chip doubles every 18 months

Bell's Law: a new computer class emerges every 10 years





Today: 1 million transistors per $

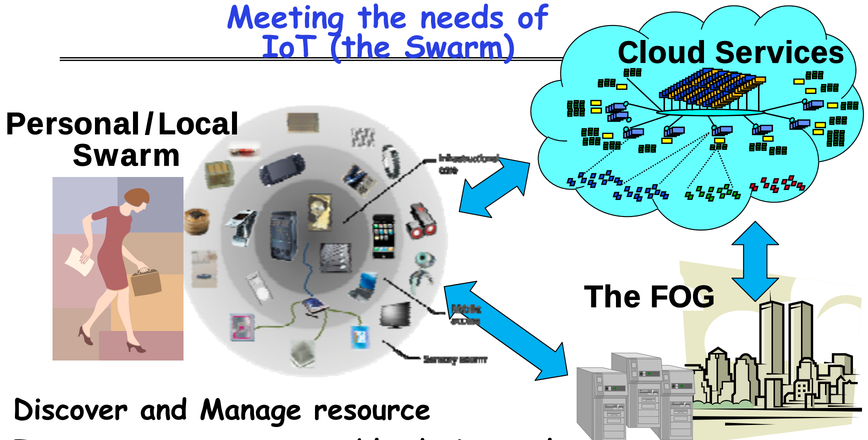
Same fabrication technology provides CMOS radios for communication and micro-sensors

'Low-Tech' Enabling Technology

Microcontroller Flash Storage Radio Communication Sensors

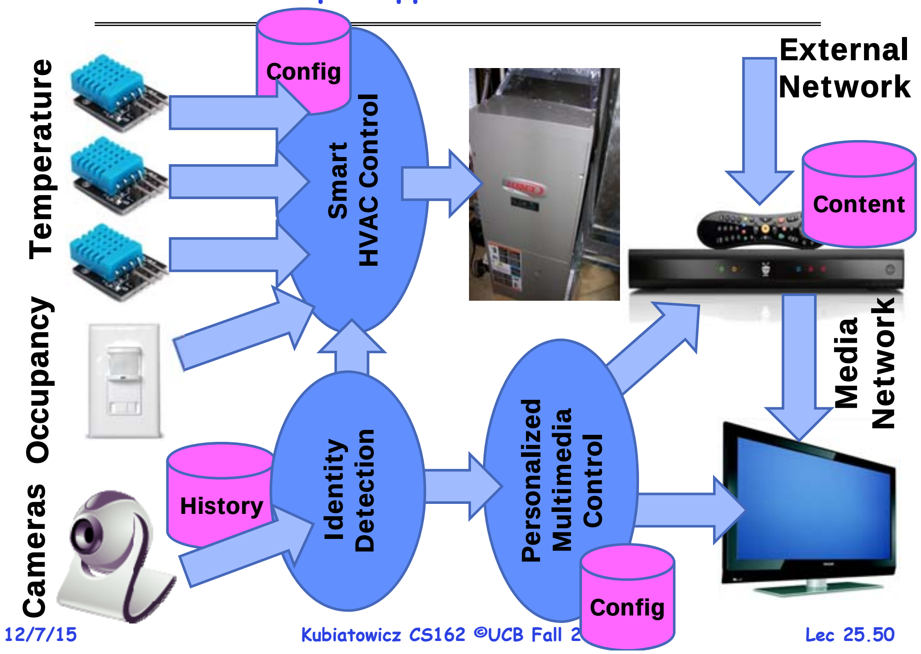
IEEE 802.15.4

Meeting the needs of IoT (the Swarm)

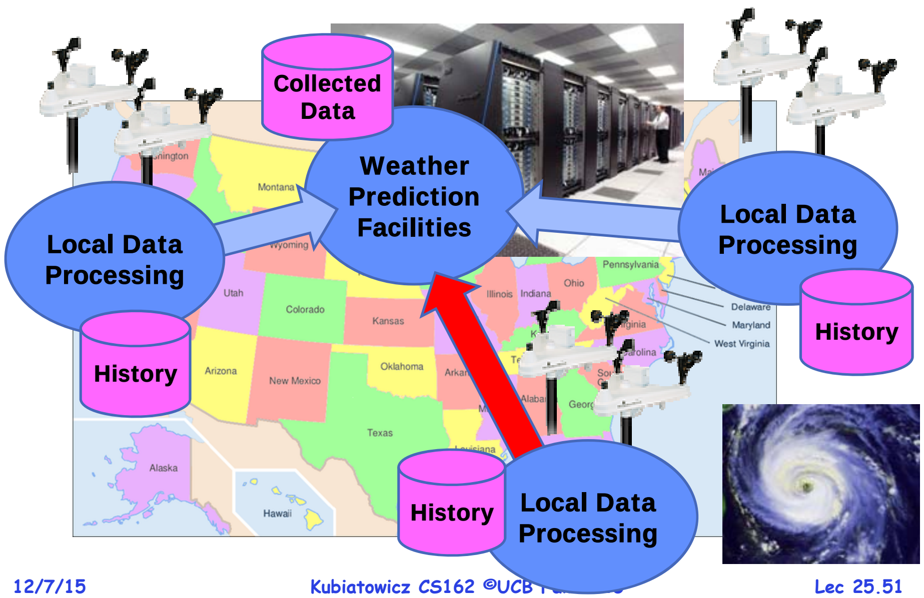


- Discover and Manage resource
- Integrate sensors, portable devices, cloud components
- Guarantee responsiveness, real-time behavior, throughput
- Self-adapt to failure and provide performance predictability
- Secure, high-performance, durable, available information
- Monetize resources when necessary: micropayments

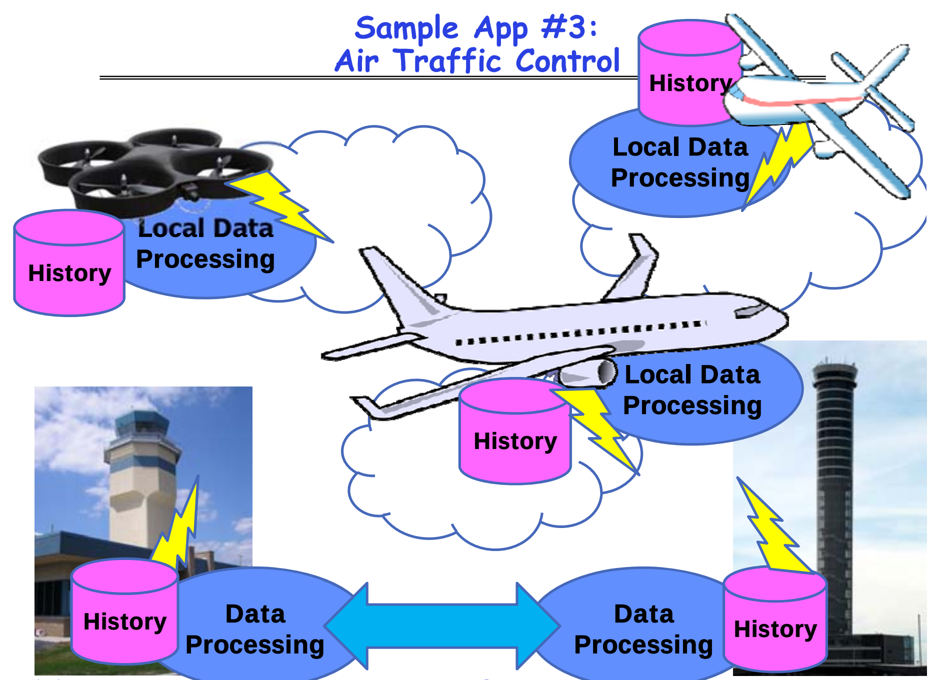
Sample App #1: Home Control



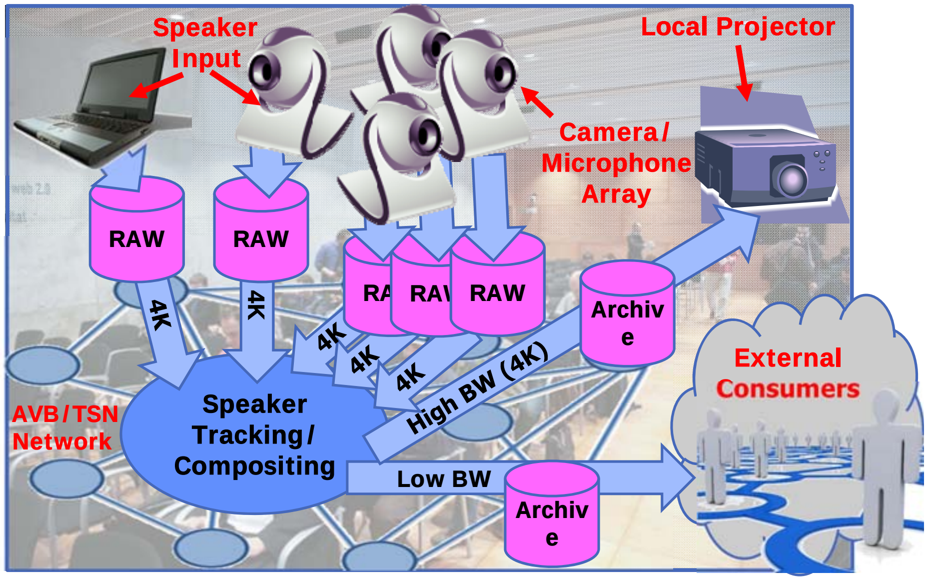
Sample App #2: Adaptive Weather Prediction



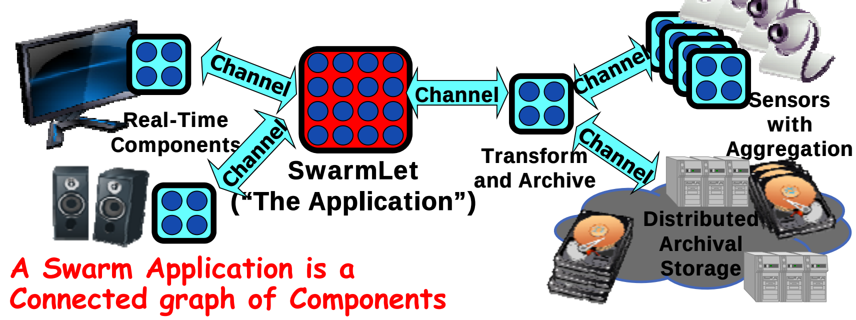
Sample App #3: Air Traffic Control



Sample App #4: Smart Seminar Room

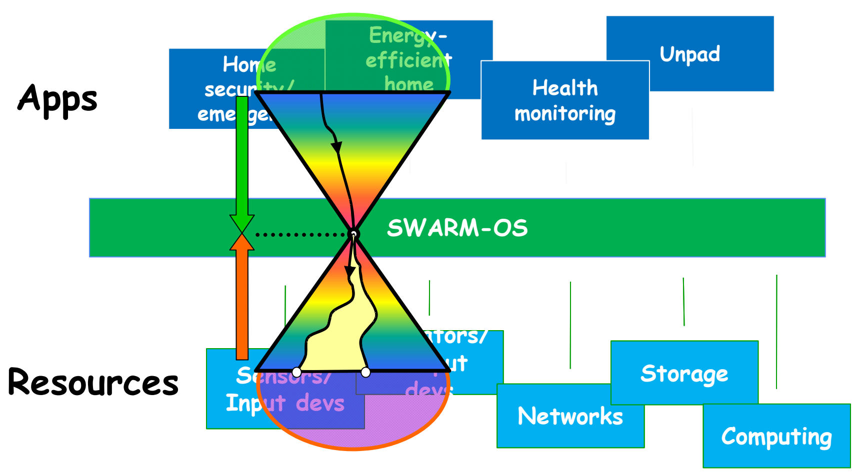


An Application Model



- A Swarm Application is a Connected graph of Components
  - Globally distributed, but locality and QoS aware
  - Avoid Stovepipe solutions through reusability
- Many components are *Shared Services* written by programmers with a variety of skill-sets and motivations
  - Well-defined semantics and a managed software version scheme
  - Service Level Agreements (SLA) with micropayments
- Many are "Swarmlets" written by *domain programmers*
  - They care *what* application does, not *how* it does it

The Missing Link?

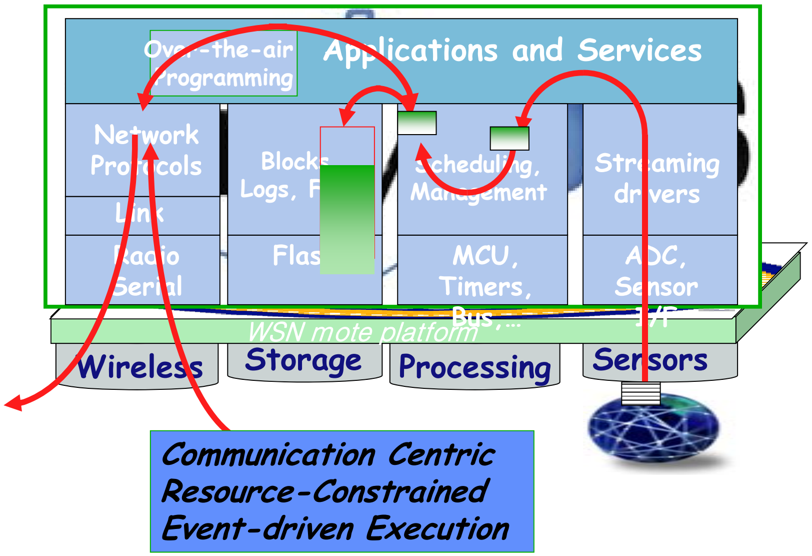


SWARM-OS: A mediation layer that discovers resources and connects them with applications

SWARMLETs

- SWARMLET: a software component written by domain programmer that is easy to write but exhibits sophisticated behavior by exploiting services distributed within the infrastructure
- Swarmlets specify their needs in terms of human-understandable requirements
  - Necessary Services, Frame rates, Minimum Bandwidths
  - Locality, Ownership, and Micropayment parameters for sensors and/or data
- Swarmlets may evolve into Shared Services
- Programmers of Services used by Swarmlets think in terms of contracts provided to Swarmlets

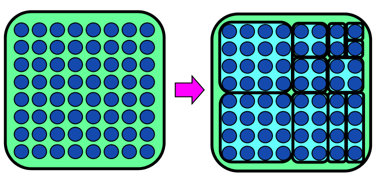
TinyOS – Framework for Innovation

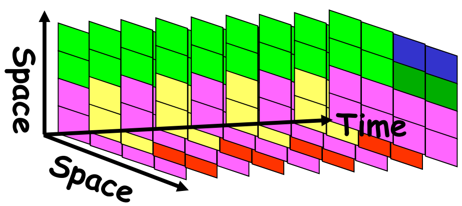


What about the "FOG" and "Cloud"? New Abstraction: the Cell

- Properties of a Cell: Service Level Guarantees
  - A user-level software component with guaranteed resources
  - Has full control over resources it owns ("Bare Metal")
  - Contains at least one memory protection domain (possibly more)
  - Contains a set of secured channel endpoints to other Cells
  - Contains a security context which may protect and decrypt information
- When mapped to the hardware, a cell gets:
  - Gang-schedule hardware thread resources ("Harts")
  - Guaranteed fractions of other physical resources
    - » Physical Pages (DRAM), Cache partitions, memory bandwidth, power
  - Guaranteed fractions of system services
- Predictability of performance ⇒
  - Ability to model performance vs resources
  - Ability for user-level schedulers to better provide QoS

Implementing Cells: Space-Time Partitioning



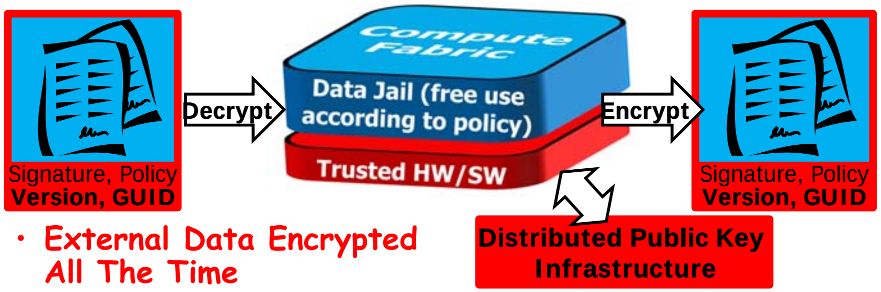


- Spatial Partition: Performance isolation
  - Each partition receives a vector of basic resources
    - » A number HW threads
    - » Chunk of physical memory
    - » A portion of shared cache
    - » A fraction of memory BW
    - » Shared fractions of services
- Partitioning varies over time
  - Fine-grained multiplexing and guarantee of resources
    - » Resources are gang-scheduled
- Controlled multiplexing, not uncontrolled virtualization
- Partitioning adapted to the system's needs

Cell Implementation Platform: Tessellation Version 2

- Tessellation Operating System
  - Provides basic Cell Implementation
  - Build on the Xen Hypervisor
- Why Xen?
  - Provides clean starting point for resource containers
  - Leverage mature OS (Linux) device support, critical drivers can be isolated in a stub domain
  - Framework for developing VM schedulers
  - Mini-OS, a lightweight POSIX-compatible Xen guest OS, is basis for the customizable app runtime
  - Support for ARM and x86
- Unikernels: Software Appliances
  - Small compiled kernels with only enough components to support one application
  - Every component has its own resource container
- Dynamic resource optimization framework

Trusted Swarm Platform

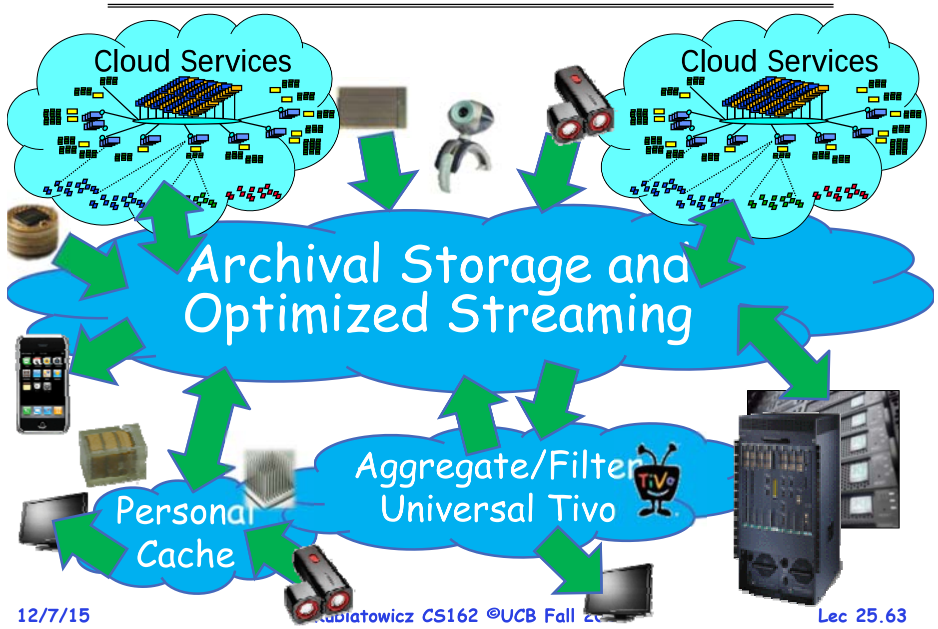


- External Data Encrypted All The Time
- Only decrypted in "Data Jails" (trusted platform)
  - Build in hardware or in software with secure attestation
  - Data leaving cell automatically reencrypted
- Trusted Platform given keys to do its work
  - Keys never given out to application software
- Built through secure boot mechanisms (i.e. TPM/etc)

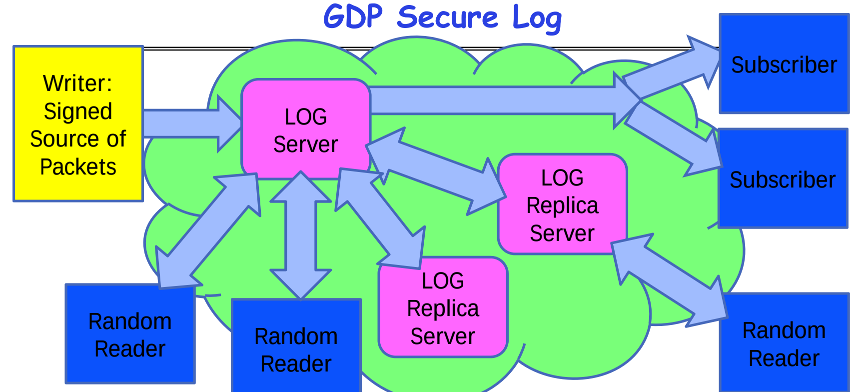
DataCentric Vision

- Hardware resources are a commodity
  - Computation resource fails? Get another
  - Sensor fails? Find another
  - Change your location? Find new resources
- All that really matters is the information
  - Integrity, Privacy, Availability, Durability
  - Hardware to prevent accidental information leakage
- Permanent state handled by Universal Data Storage, Distribution, and Archiving
- We need a new Internet for the Internet of Things?
  - Communication and Storage are really duals
  - Why separate them?

The Global Data Plane

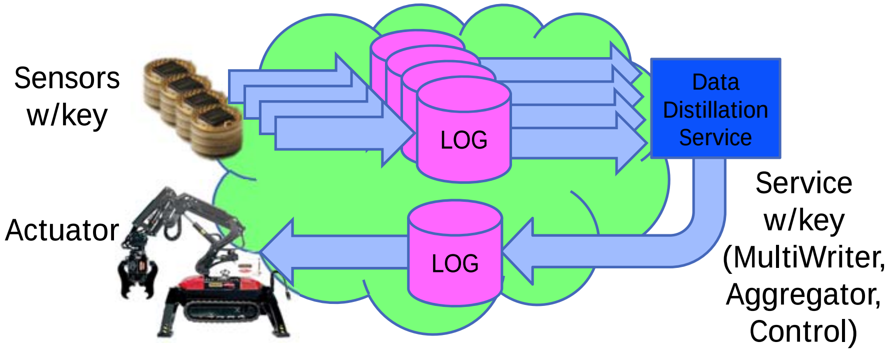


GDP Secure Log



- Locality Optimization/QoS
  - Flat 256-bit address space
  - Routes adapted as elements move
  - Hardware QoS exploited
  - Multicast trees built as needed
- Durability
  - Replicas/Reed-Solomon coding
- Single Writer/Append only
  - Owner Key Signs entries
  - LOG server rejects bad entries
  - Tradeoff in granularity, i.e. frequency of signatures
- Multiple Readers/Subscribers
  - Random access/push based

Simple Log-based Use-case



- Lightweight Logs ⇒ One Log per Device
- Log Input Secured via Owner Key/Checked by consumer
- Optional encryption for privacy
- Timestamps to help ensure freshness

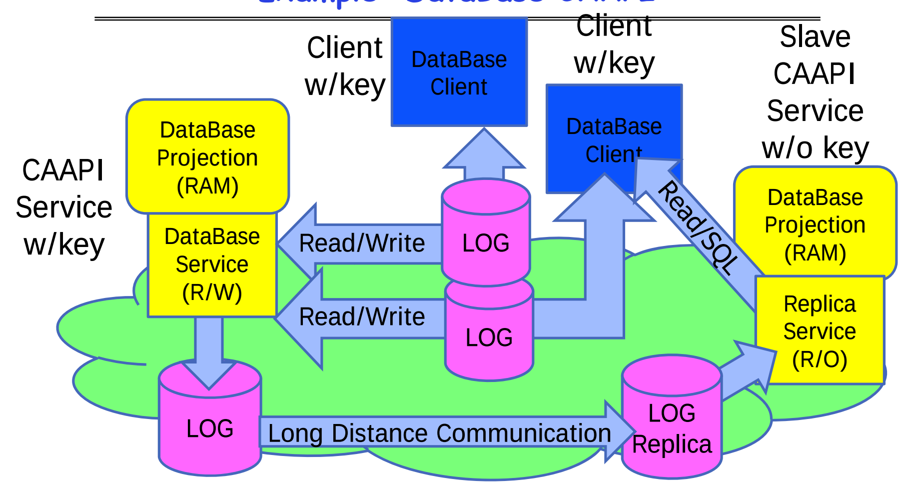
Security of Data in Log

- Two Completely Different uses of Keys:
  - Each LOG associated with an Owner public key
    - » This key must be known *and protected* by writers
    - » All data in GDP must be signed ⇒ Single writer log
  - Each LOG associated with one or more keys for Encryption
    - » The use of encryption keys not mandated by GDP infrastructure
    - » However, will (shortly) have sample/recommended use cases and libraries to support different styles of encryption
- Writer has sole control over integrity of data
  - LOG servers may deny existence of data, but can not forge it
  - Public key of writer established at LOG creation time
  - Next version of GDP will have authentication support
- Automatic construction and registration of LOGs
  - New sensors tied into GDP via secure registration process
  - Ultimately, interaction with Control Plane

Build DataStores on top of GDP through Composition

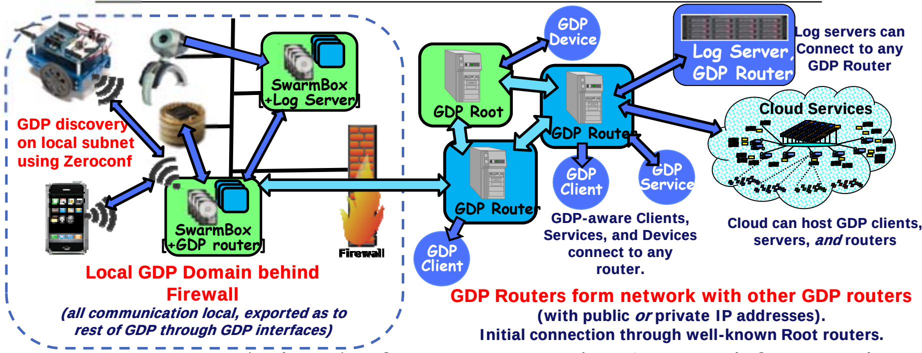
- Common Access APIs (CAAPIs): Support common data access methods such as:
  - Key/Value Store
  - Object Store/File System
  - Data Base (i.e. Google Spanner)
- CAPPIs exported by services that consume the LOG
  - Much more convenient way to access data
- The LOG is the *Ground Truth* for data, but data is projected into a more convenient form
  - To do Random File access, Indexing, SQL queries, Latest value for given Key, etc
  - Optional Checkpoints stored for quick restart/cloning

Example: DataBase CAAPI



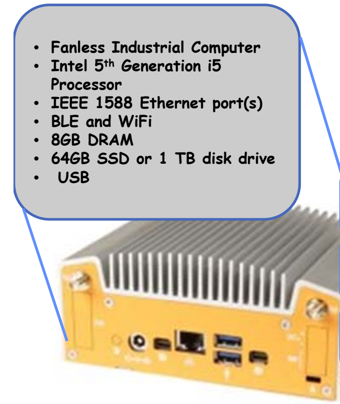
- CAAPI Service can be taken down, replicated, and restarted
- Time-stamp driven transactions (Google Spanner)
- Cloud-based computation (Spark)

The Global Data Plane (Physical View and Status)



- Current status: deployed infrastructure with substantial functionality
  - GDP router works behind firewalls and preserves locality
  - Tolerates failure and incremental addition of new routers
  - Log servers check data signatures and push updates to subscribers
  - Discovery GDP infrastructure through combination of Zeroconf protocol (same as Apple Bonjour) and/or connection with well-known GDP Root Routers
  - Delivery of data through push-based multicast (works across subnets)
  - SwarmBox can serve as GDP Router, Log Server, and Client

SwarmBox (HW and SW)



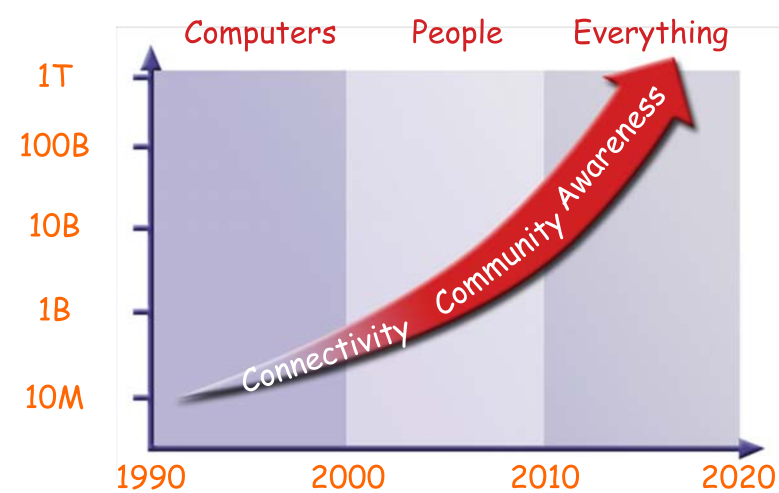
Applications	Proactive Environments
Swarm Services	Localization, Identification, Gestures Context Modeling, Machine Learning
SwarmOS Services	Brokerage, Computation, Hosting, Service Proxying, Service Launch/Monitor/Teardown, Time Access, Device/Service Discovery
Global Data Plane	Communications, Routing, Data Storage
COTS System Software	Linux, DHCP, Firewall, IP Router Container/VM, 802.1AS
SwarmBox Hardware	Intel NUC, i210 enet, Beagle Bone Black

- Current status: Version 2 now available
  - Pre-loaded with a variety of Software Components
  - Management through Docker and Ansible
  - Operating version of the GDP Router and Log Server
- Complete list of current boxes available at: https://www.terraswarm.org/testbeds/wiki/Main/SwarmBoxes

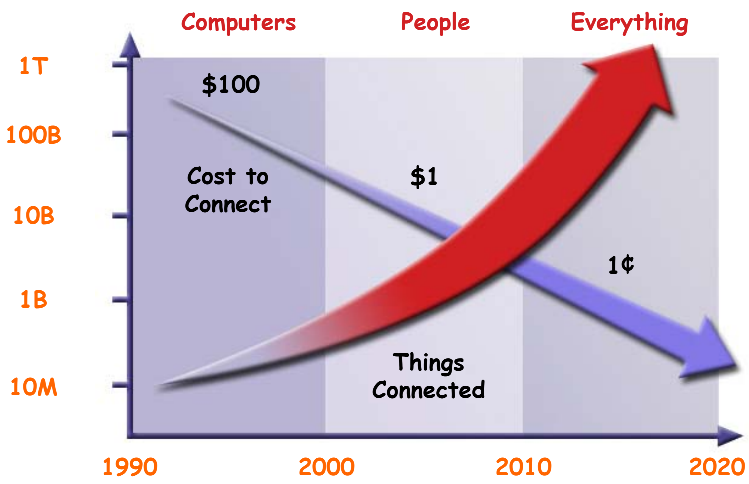
Properties of the GDP (Summary)

- Universal way to address every stream of information
  - Publish/Subscribe view of information
  - Large *flat* address space (at least 256 bits)
  - Mechanisms for access control, privacy, and transactions
  - Streams of data persisted automatically for later access
- Location Independence ⇒ Above network level
  - Build Swarmlets once and run them anywhere
  - Migrate or replicate *running* swarmlets
  - Locality optimization/QoS handled by underlying system
- Common Access APIs (CAAPIs) provide standard Interfaces
  - Key/Value Store, Data Bases, File Systems
- Deep Archival Storage:
  - Automatic Geographically Distributed Archival Storage
- One system for sensors and big data

The Revolution



The Revolution



Good Luck on the Final!

- Quantum Computing
  - Shor's Factoring Algorithm: Factor large numbers in polynomial time
  - Ion Traps provide potential to scale with Moore's law
  - Quantum CAD: Optimize limited resources
    - » Makes particular sense for Quantum Computers!
- Internet of Everyday Things
  - Soon – everything connected all the time
  - Wireless, Wired, FOG and Cloud
  - How to build a useful infrastructure for the future?
    - » Computation everywhere
    - » Global Data Plane: Truly ubiquitous storage
    - » Applications that span many services and geographical distances