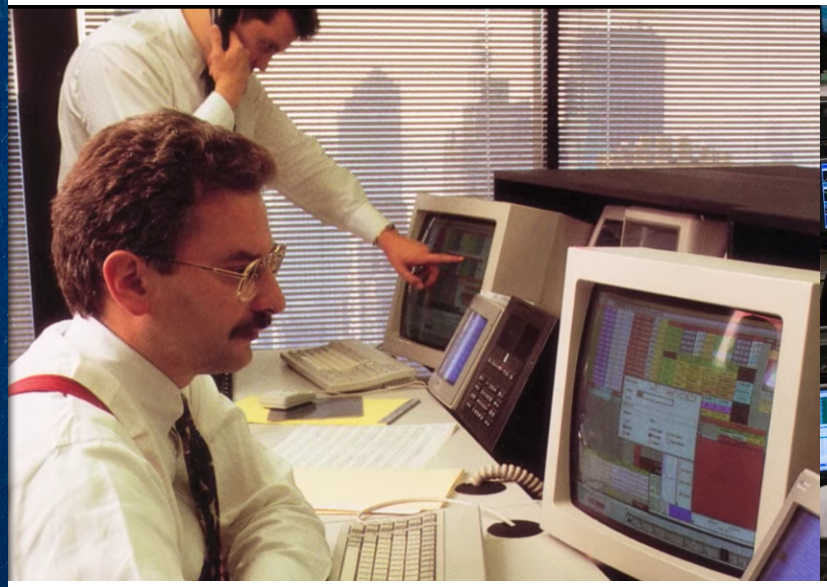
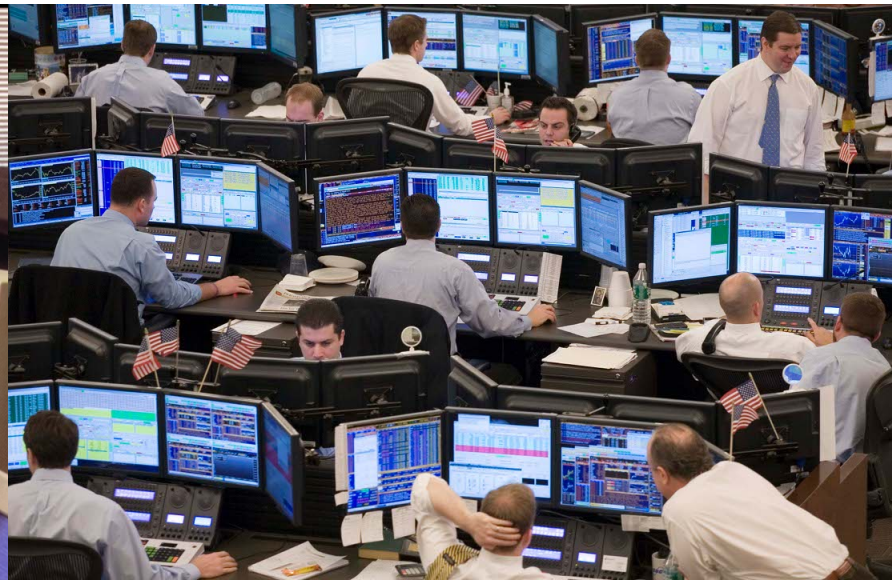
Middleware's Presentism: Asynchrony, Flow, Finance, and the Enterprise





Michael Castelle
University of Warwick
HaPoP 2018, Oxford, UK
23rd March 2018

Outline

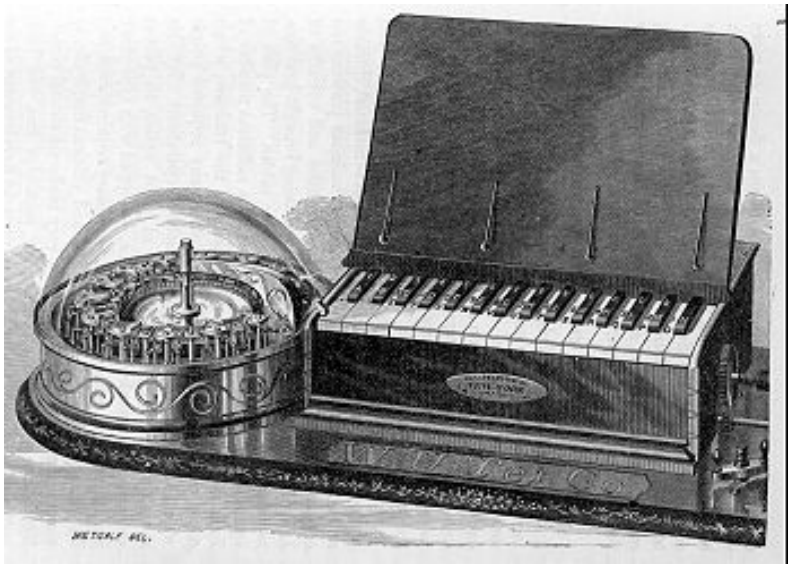
- Data vs. Data-in-motion
- Synchrony vs. asynchrony in the history of telecommunication
- Historical genealogy of "publish-subscribe"/multicast message broker systems in financial services
- Implications of data-in-motion/ asynchrony/multicast qualities for history/philosophy of programming and (distributed) computing

Data
(static; finite)

Data-in-Motion
(dynamic; potentially infinite)

Wall Street: a 150 Year Tradition of Data-in-Motion

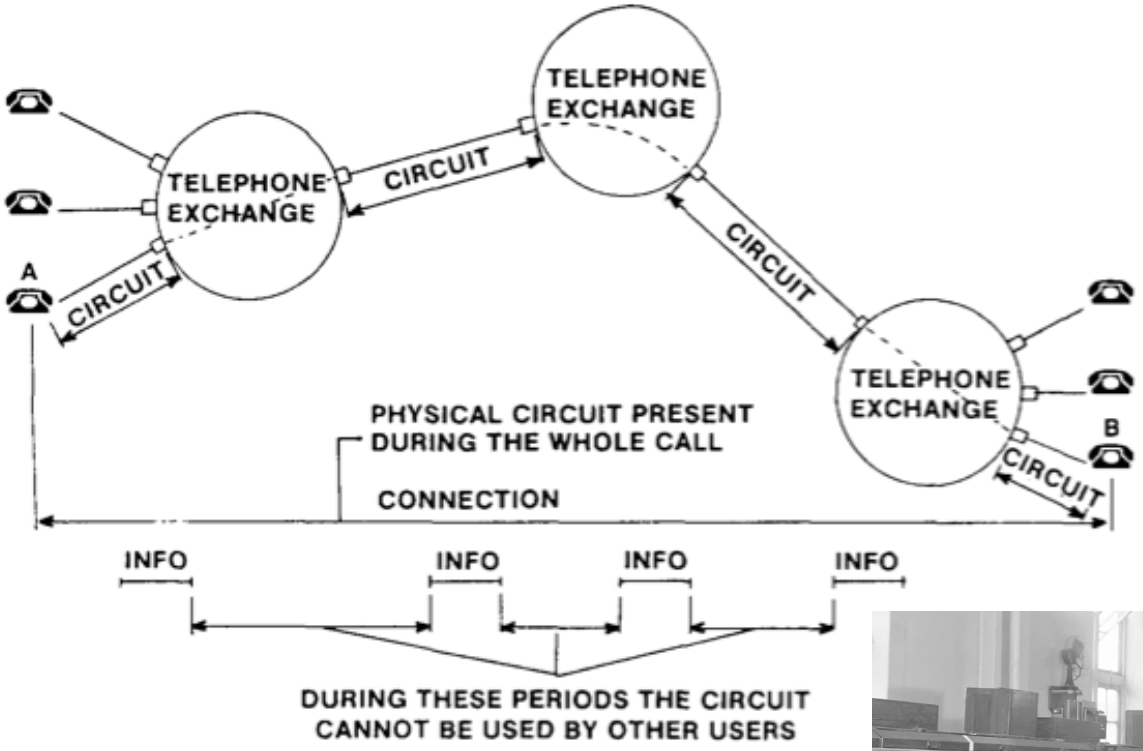
- Stock ticker
  - NYSE (1867)
  - London Stock Exchange (1872)



Knorr Cetina and Preda (2007) “The Temporalization of Financial Markets: From Network to Flow”

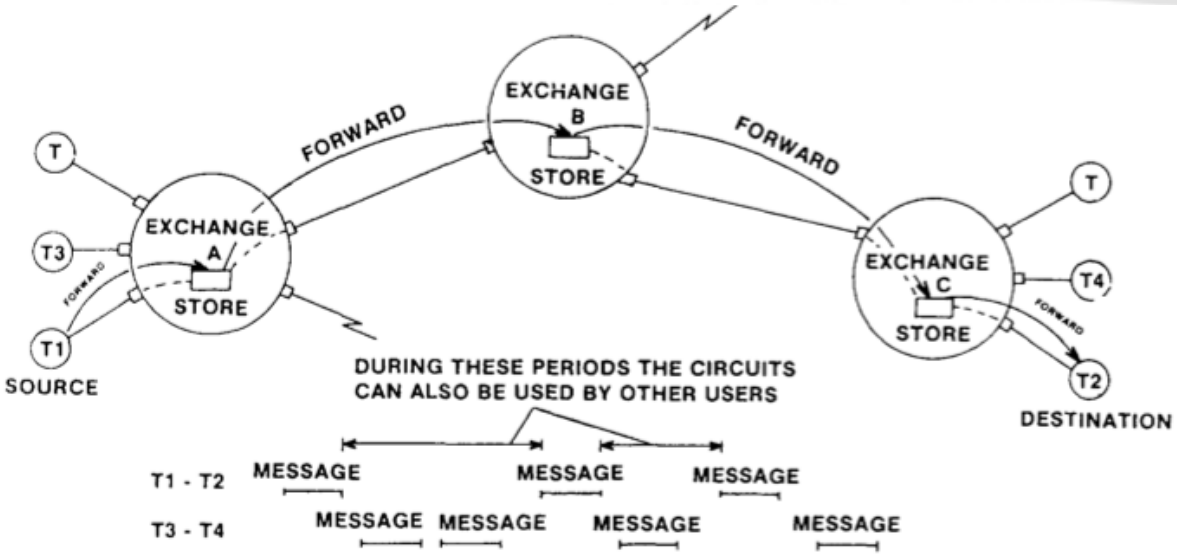
Paradigms of Telecommunication: *Line Switching*

(Indexical/Synchronous)



Paradigms of Telecommunication: *Message Switching* / "Store-and-Forward"

(Symbolic/Asynchronous)



“Torn Tape” Relay in the Vietnam War

U.S. Signal Corps mobile communications center

“Torn Tape” Relay in the Vietnam War

U.S. Army Teletype Center in Phu Lam, Vietnam

Message Glut on Wall Street

New Ticker to Aid Stock Market's Heartbeat

BY ROBERT SULLIVAN, Times Assistant Financial Editor

A LOG-JAM in securities trading is about to be broken.

The New York Stock Exchange's nationwide network of transmission equipment, which has been sorely taxed to keep up with the increasing volume of trading, is being overhauled. New high-speed tickers will almost double the capacity to report the latest quotation of each trade made.

Within the next few weeks—hopefully, next month—Western Union will complete installation of hundreds of the new tickers in brokerage houses and stock exchanges that subscribe to the NYSE's quotation service.

It is a gigantic task. The job involves installation of some 875 of the new "900" tickers in downtown New York City alone. Nation-wide about 2,500 old tickers are being replaced with the new. The changeover started last July.

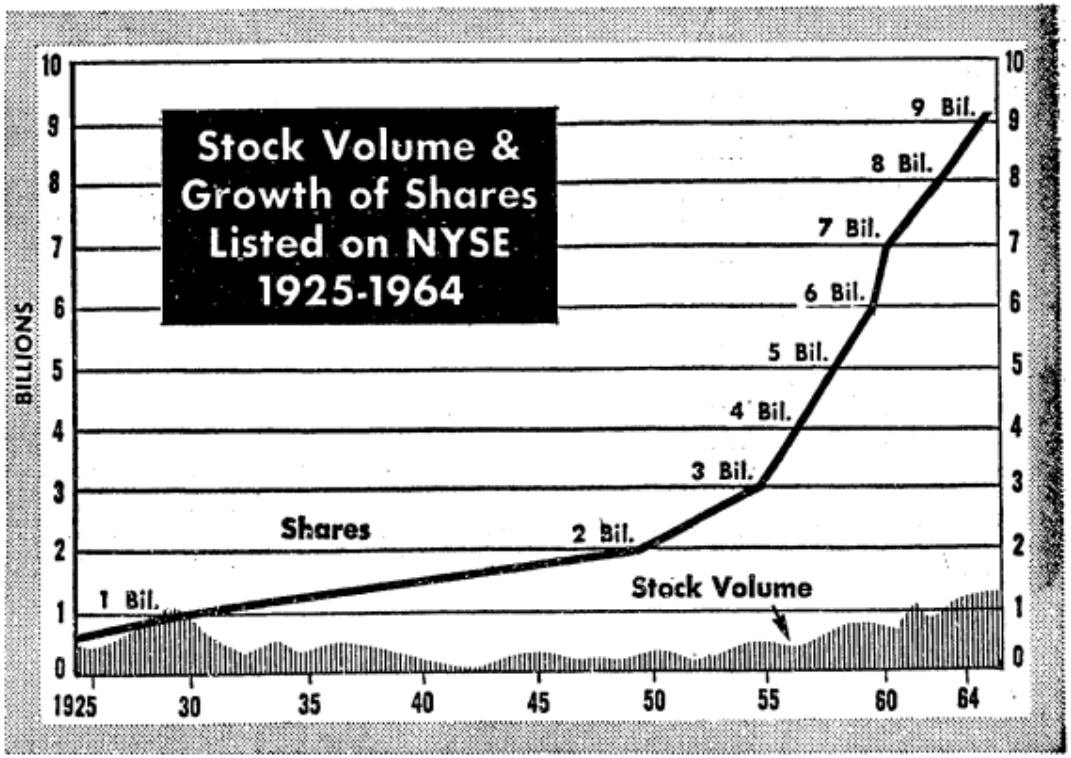
The sleek "900" ticker, developed for the NYSE by Teletype Corp., gets its name from the fact that it can rattle off 900 characters a minute when operating at top speed. The best the old units can do is 500.

The new network of tickers is expected to eliminate all time lag in reporting transactions made on the NYSE in periods of worst possible stress.

With increasing frequency in the past, waves of heavy trading have caused delays of from a few minutes to a few hours in transmitting the volume and price of each individual order executed on the floor of the NYSE. As a result, in such periods of hectic trading, investors and brokers around the country have been in the dark as to what price they could buy or sell a security. It has been a frustration in the securities market that shortly will be eliminated.

Brokers are breathing easier now that the new system is almost a reality. Trading volume is pyramiding. It is already well on its

Please Turn to Pg. 8, Col. 5



STEEP CLIMB—It took 137 years for NYSE to list first billion shares, less than nine months for ninth billion. Times chart

Message Glut on Wall Street

Big Board's Fast Ticker Straining to Keep Pace

By VARTANIG G. VARTAN

"Fantastic." "Amazing." "Unbelievable."

On Wall Street, where superlatives usually are reserved for great events, stockbrokers are using these words to describe the trading volume in April on the New York Stock Exchange.

So far this month, turnover at the Big Board has averaged a shade higher than 15 million shares a day. Truly fantastic, amazing and unbelievable when one considers that 15 years ago, when the postwar bull market was well under way, volume averaged merely 1.4 million shares a day—or less than one-tenth the rate in April, 1968.

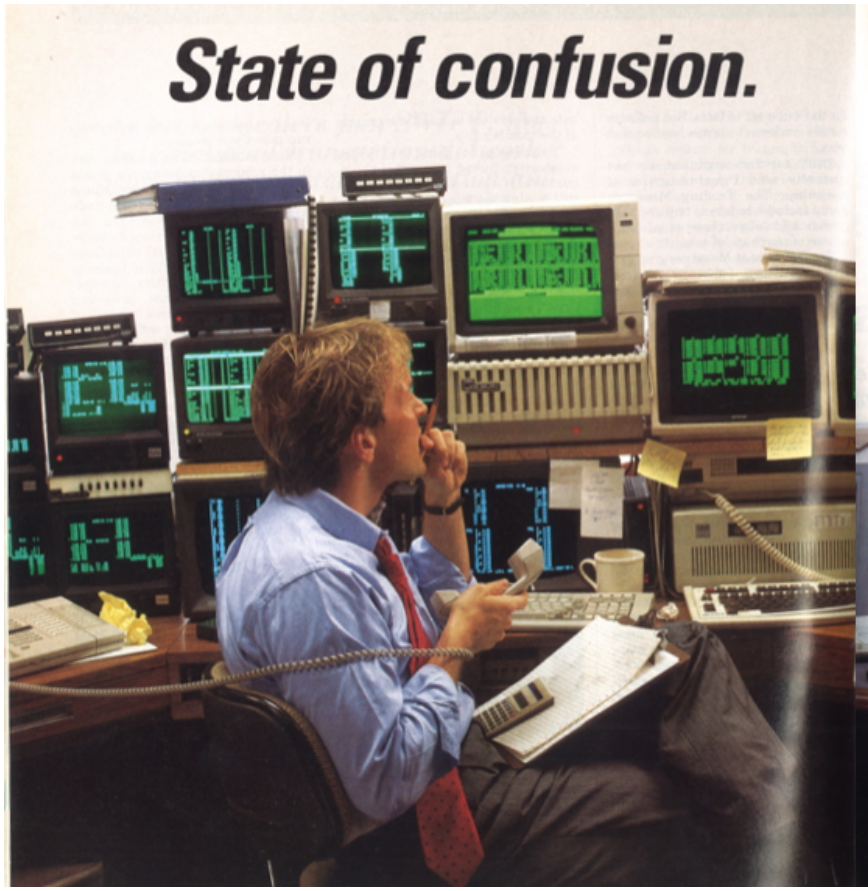
The huge volume currently enjoyed by the exchange places enormous pressures upon the "900" ticker placed in service in late 1964 "to inaugurate a new era in the reporting of stock-market information."

Brokers and investors agreed last week that the doughty ticker, despite its lag behind floor transactions in periods of peak activity, has been performing with remarkable effectiveness.

Meanwhile, behind the stately Italian Renaissance facade of the exchange, efforts are being intensified to improve the present ticker for high-volume activity and

Continued on Page 10, Col. 3

Mid-1980s Wall Street: the Data Deluge



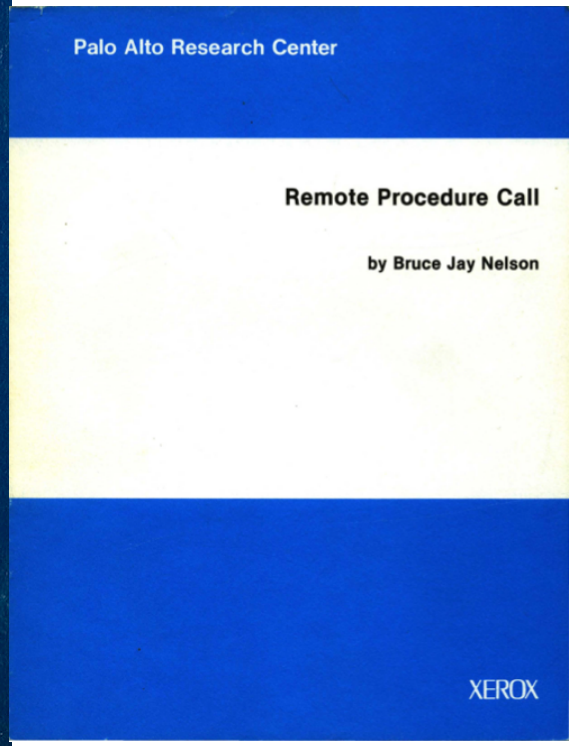
Micrognosis advertisement, *Wall Street & Technology,* 1989

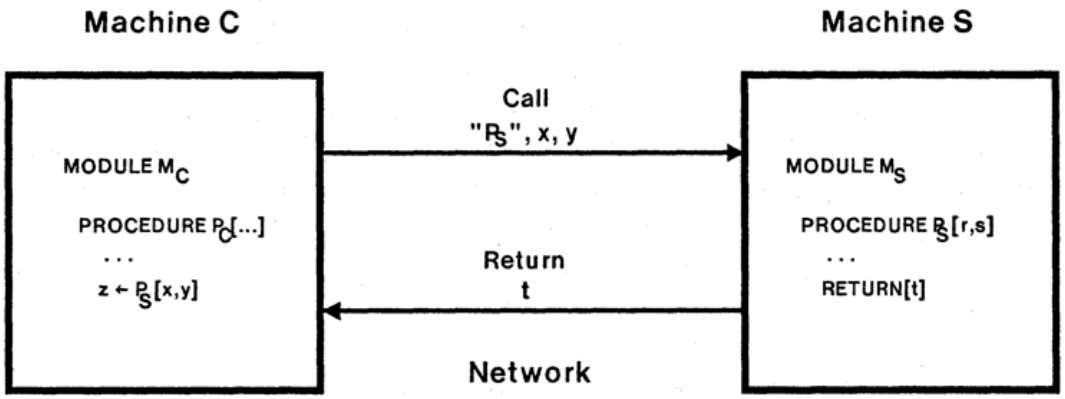
Mid-1980s Wall Street: the Data Deluge

Wall Street (1987), 20th Century Fox

Communication in Early Distributed Systems: *Remote Procedure Call (RPC)* (1981)

(synchronous; request/reply)





From Nelson (1981) “Remote Procedure Call” (Xerox PARC Report). A diagram of a remote procedure call (RPC) from client (“Machine C”) to server (“Machine S”).

Group Communication: The V Kernel (1984) (synchronous; multicast)

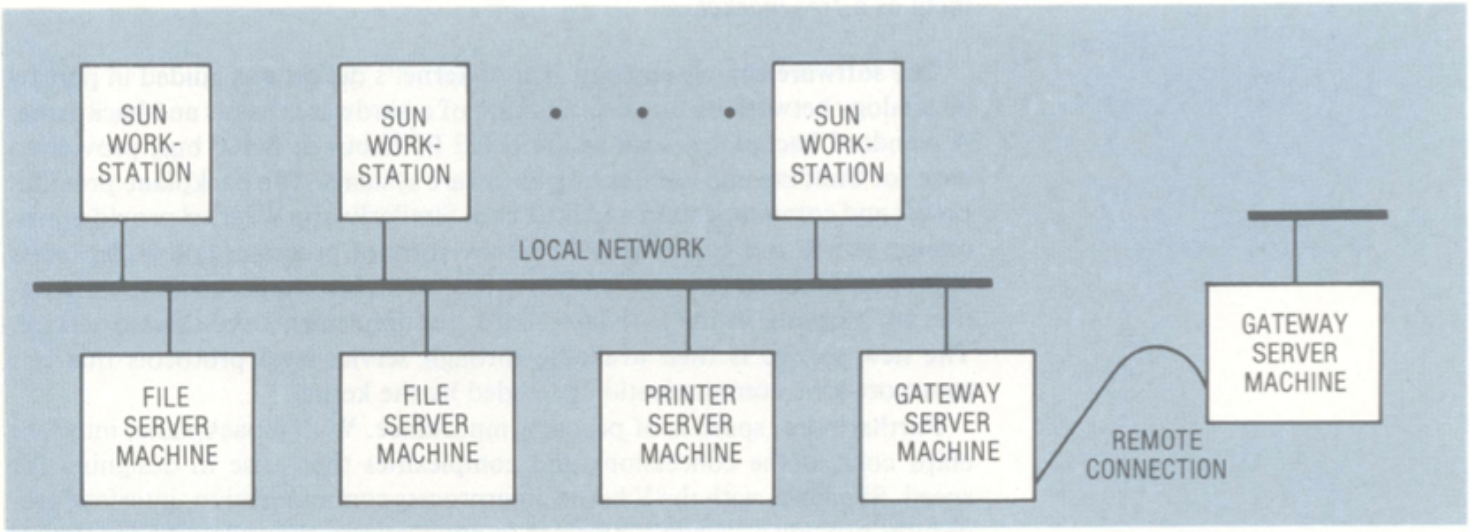


Figure 1. A V domain of local network-connected machines.

- "An attractive paradigm for a distributed system is that of a *free marketplace*. Services are offered by servers, while clients communicate with servers to negotiate and receive services… In contrast to this free market model, a single machine operating system acts as a *centrally planned economy*. All hardware resources are controlled and allocated *by a benign dictator* that provides services to applications…"

Cheriton (1984) "The V Kernel: A Software Base for Distributed Systems"

Group Communication: The V Kernel (con't.) (synchronous; multicast)

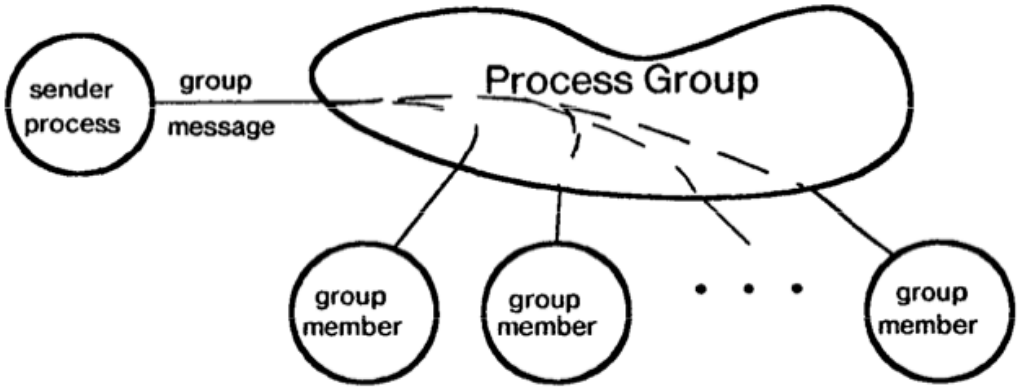
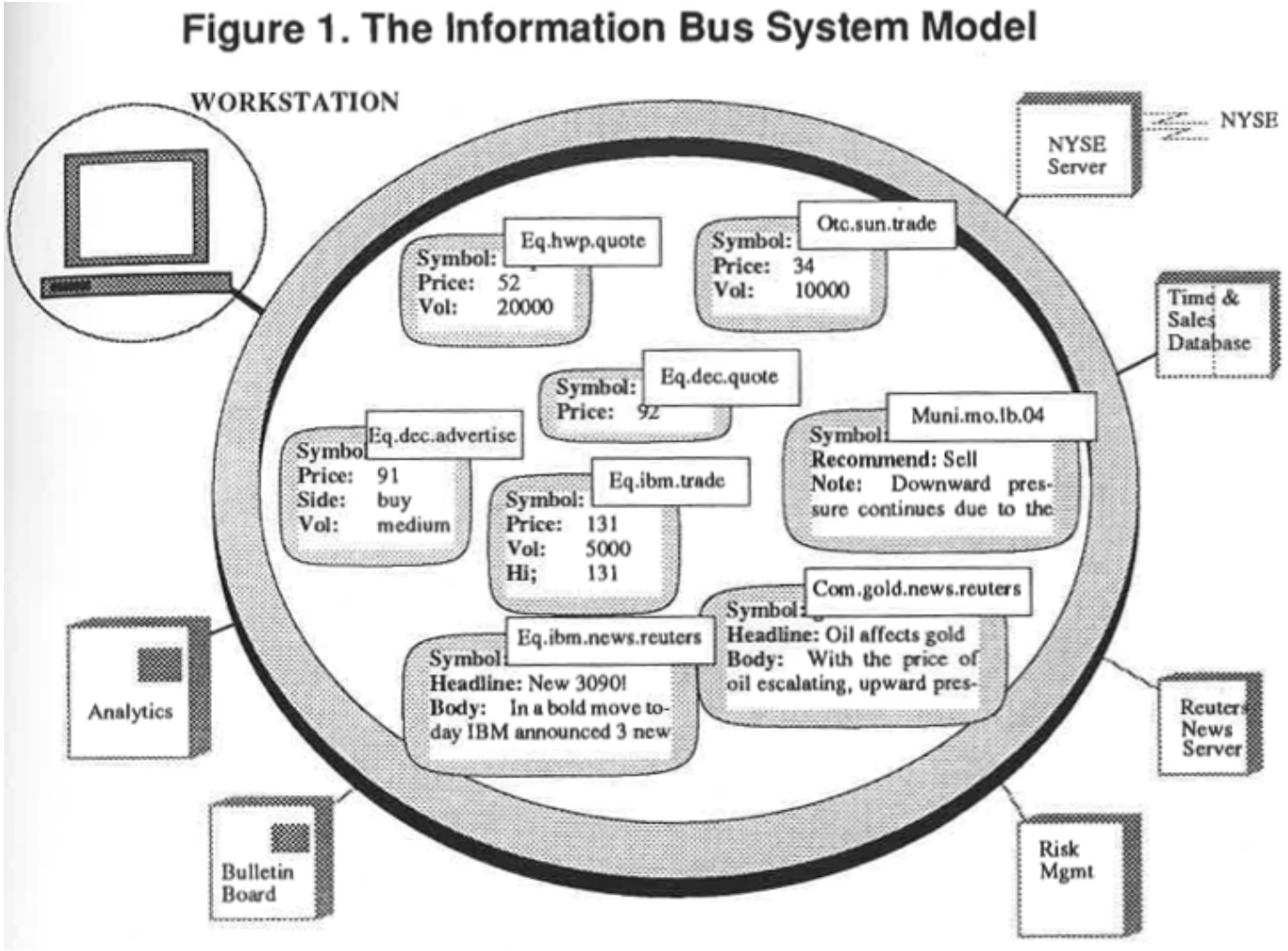


Fig. 5. Group message forwarded to group members.

Putting the onus on the receiver for reliable delivery leads to what we call *publishing*.[10] It is so named because it mimics real world publishing. That is, information to be sent to a group, the *subscribers*, is filtered through the *publisher*, which collates and numbers the information before issuing it to the subscribers.

Cheriton and Zwaenepoel (1985) "Distributed Process Groups in the V Kernel", *ACM Transactions on Computer Systems*

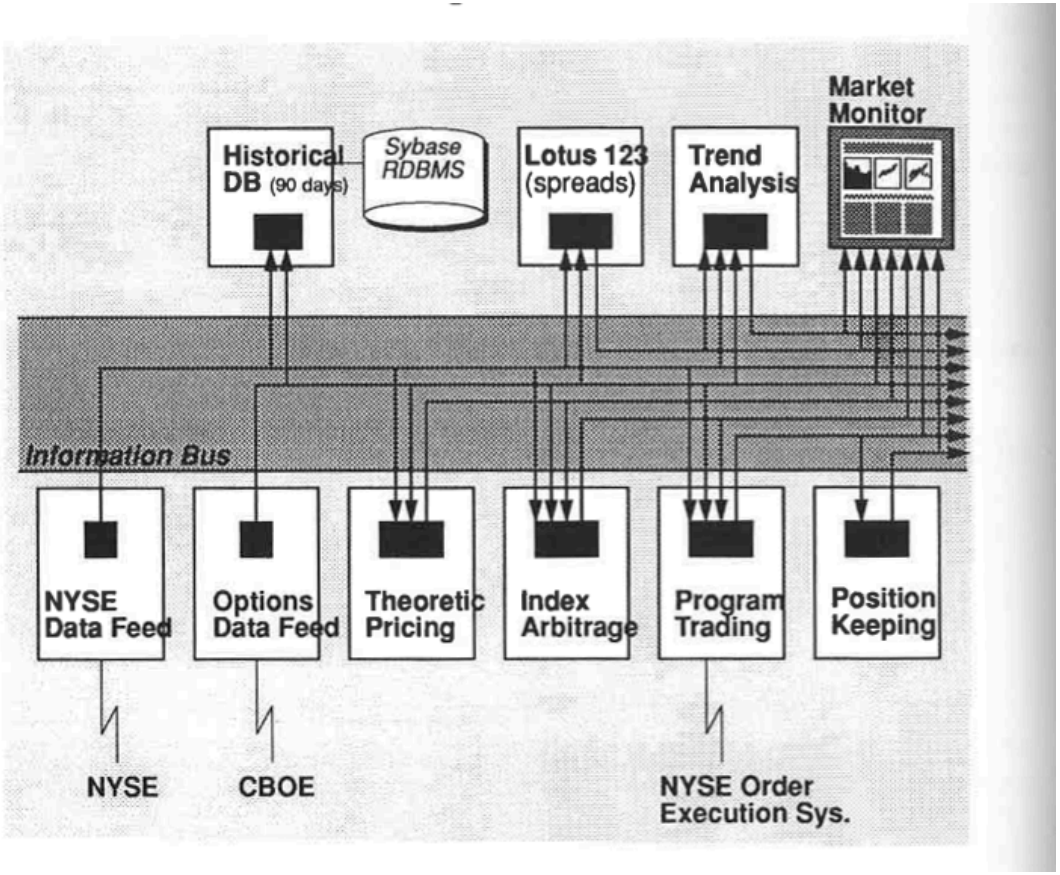
The Information Bus (TIB) (1986)
(asynchronous, multicast)



- Seeded by Teknekron Corporation, early startup incubator (1968, Berkeley)
- 1987: Spun off into independent company

Skeen (1992) “An Information Bus Architecture for Large-Scale, Decision-Support Environments” (Winter Usenix Conference ’92, San Francisco)

The Information Bus (TIB) (con't.)
(asynchronous, multicast)



- "This paper concentrates on the problems posed by a "24 / 7" commercial environment…"
- "To disseminate data objects, data producers generate them, label them with an appropriate subject, and *publish* them on the Information Bus… data *consumers* subscribe to the same subject. Consumers need not know who produces the objects, and producers need not know who consumes or processes the objects."

Oki, Pfluegl, Siegel and Skeen (1993) "The Information Bus®—An Architecture for Extensible Distributed Systems", *SIGOPS '93*

TIB / Sun Microsystems on Wall Street (1988)

Schmerken, “To Build or to Buy?”, *Wall Street & Technology*, January 1989

IBM: MQSeries (1993-)

- Developed at IBM Hursley (home of the CICS transaction monitor)
- Early 1990s: IBM partners with Systems Strategies Inc. (ezBridge Transact)
- 1993:
  - Defines Message Queue Interface (MQI) API
  - Message Queue Manager for MVS/ESA released

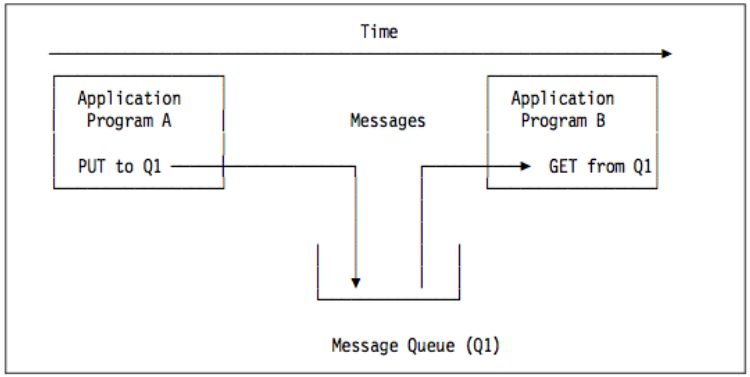


Figure 19. Message Queuing: Principle

The two main aspects of this method are:

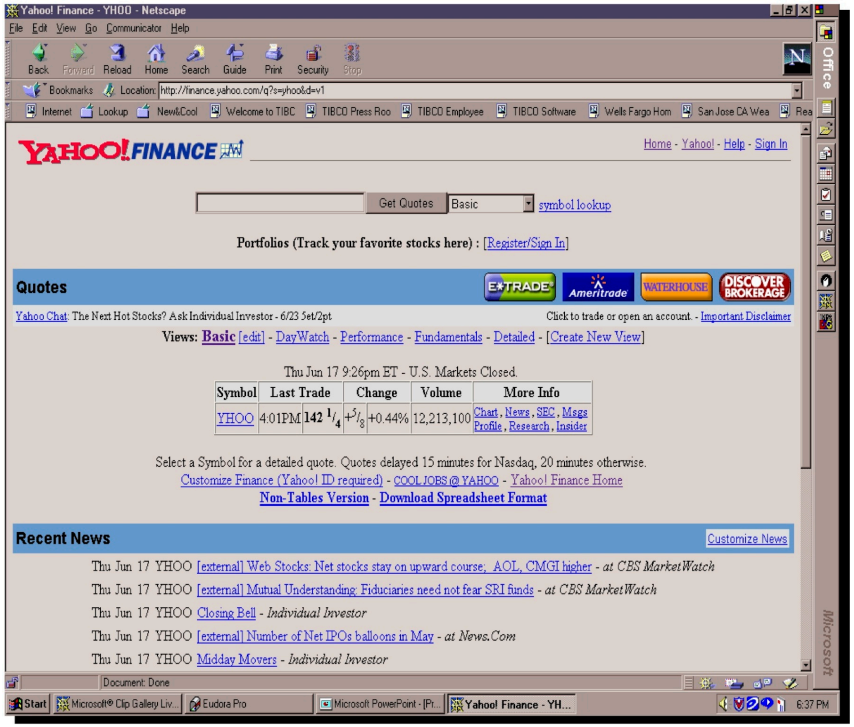
- Programs communicate through queues.
- Programs communicating through queues need not be executed concurrently.

"An Early Look at Application Considerations Involved with MQSeries", December 1994

TIBCO and the Financialization of Everyday Life

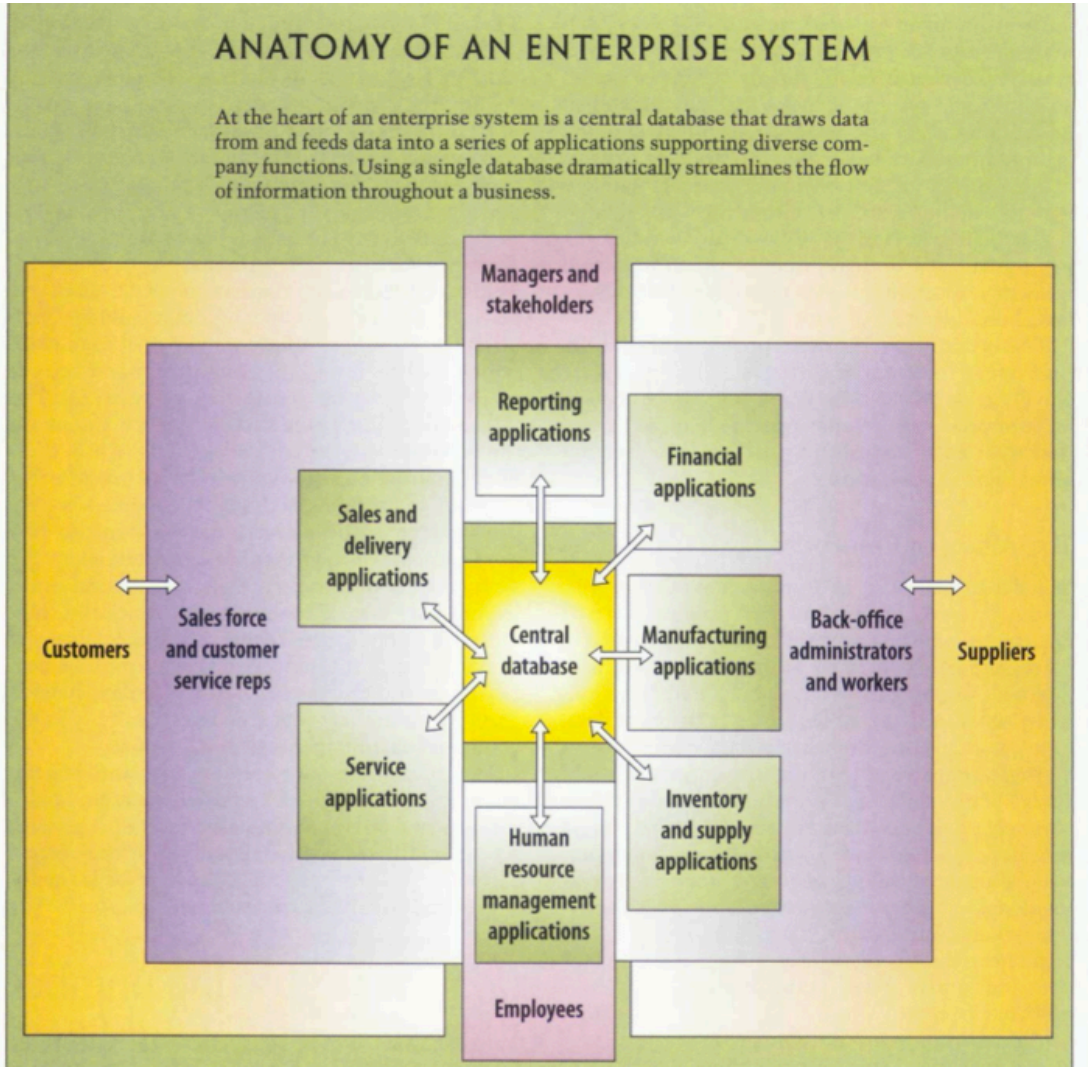
- Teknekron Software acquired by Reuters, December 1993 for $125 million
- 1996: Renamed TIBCO

TIBCO and the Financialization of Everyday Life



- Teknekron Software acquired by Reuters, December 1993 for $125 million
- 1996: Renamed TIBCO
- Late 1990s: TIB used for Yahoo! Finance backend
- 1999-2001: Partners with Altavista, Forbes, NBC to provide real-time stock quotes

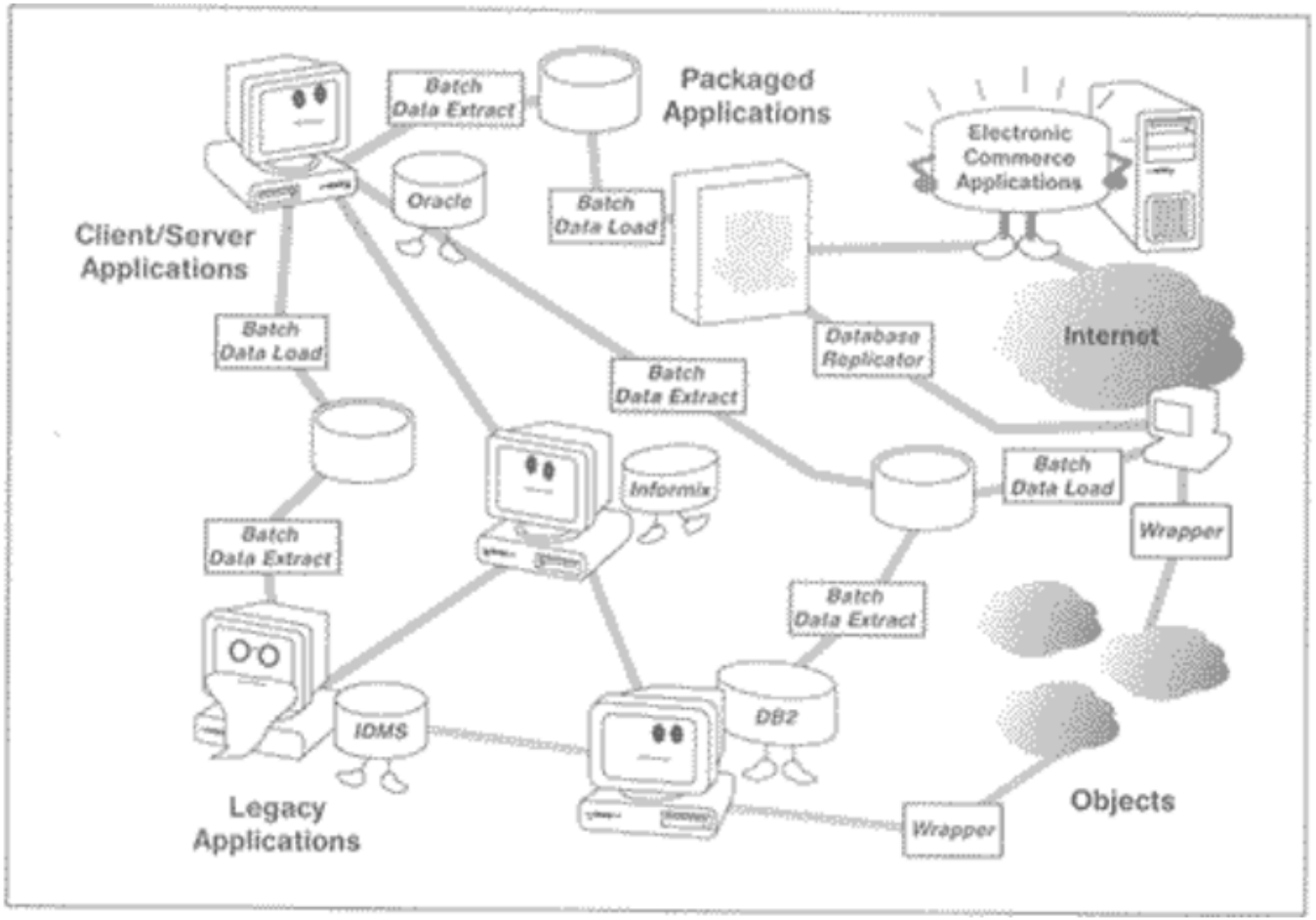
A Deceptive Portrayal of The Enterprise



Davenport (1998) "Putting the Enterprise into the Enterprise System", *Harvard Business Review*

The More Likely Reality of The Enterprise

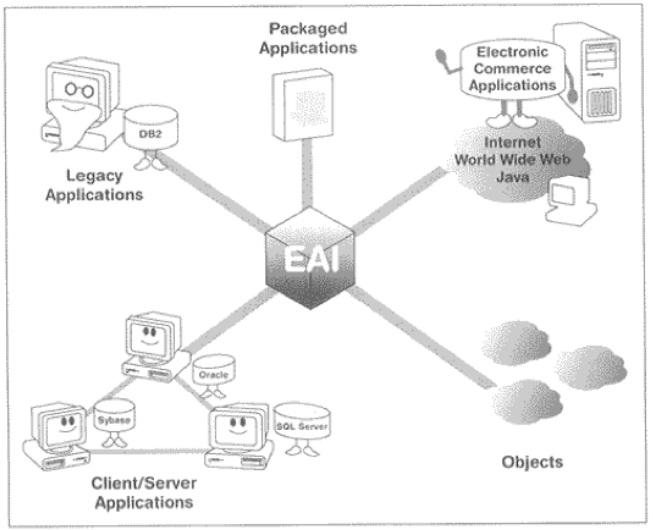
Figure 1.4 Enterprise chaos



Linthicum (1999), *Enterprise Application Integration*

Enterprise Application Integration (EAI)
(late 1990s-early 2000s)

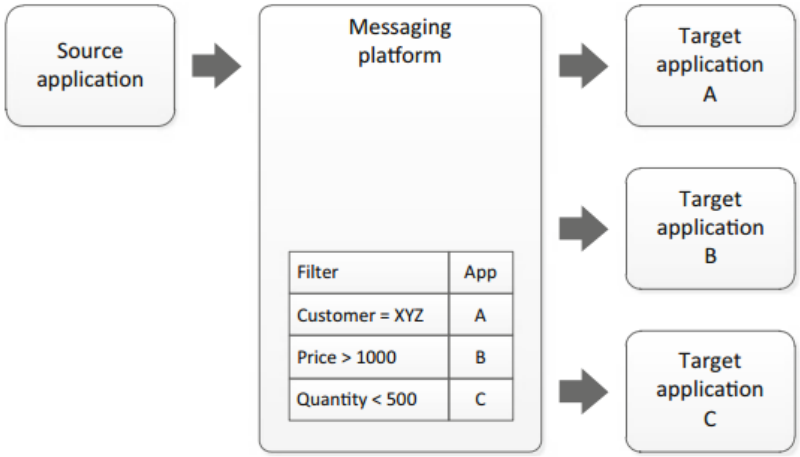
- Heterogeneity is the norm
- Geographical diversity
- Legacy platforms
- Incompatible systems
- Variety of databases



Linthicum (1999), *Enterprise Application Integration*

Goal of "message-oriented middleware" (MOM) is to better 'glue' these disparate systems together through asynchronous messaging

- IBM MQSeries Integrator
  - *content-based* message routing
- TIBCO Active/Enterprise



Middleware in Finance at the Turn of the Century: "Unheralded Yet Vital"

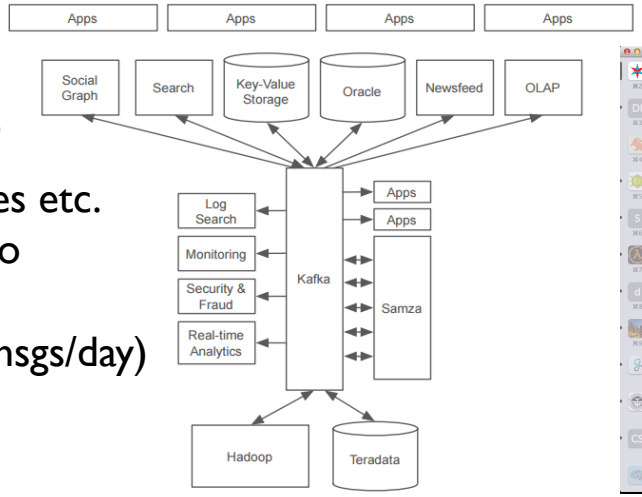
For financial firms, middleware is like oxygen—pervasive, unseen and often taken for granted, yet essential for life. Banks use it to shunt information between systems of record, customer touchpoints and the wire room. Brokerages use it to collect financial data for cleansing and distribution, translate proprietary data into standard formats (e.g., FIX, ISITC, SWIFT) and automate back-office tasks. Insurers use it to access legacy data—most recently to comply with government regulations for health insurance portability.

Middleware enables manufacturers and distributors of financial services to work together, such as a bank transmitting customer data to insurance underwriters in exchange for live policy quotes. Vendors like NEON, Tibco, BEA Systems, Financial Fusion and IBM (MQSeries), supply financial institutions with key middleware functions such as reusable objects for guaranteed message delivery, shared data definitions and transaction logging.

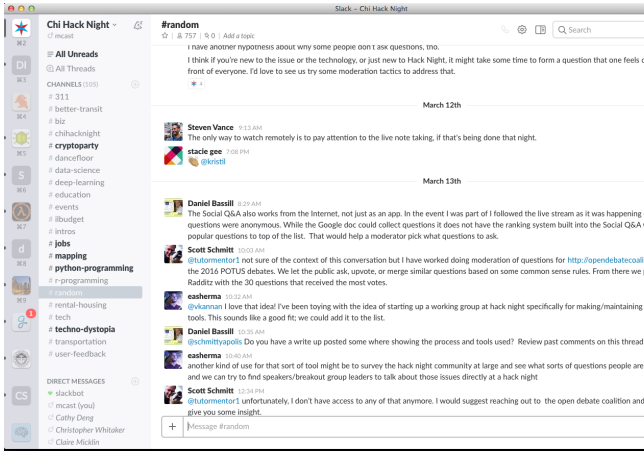
"Technology makes convergence a reality', Schmerken et. al., *Bank Systems & Technology,* May 2001

Contemporary Message Brokers in Surveillance and IoT

- **Kafka**
  (LinkedIn, 2011-)
  - Pageviews, Searches etc. each "published" to separate "topics"
    (near one billion msgs/day)

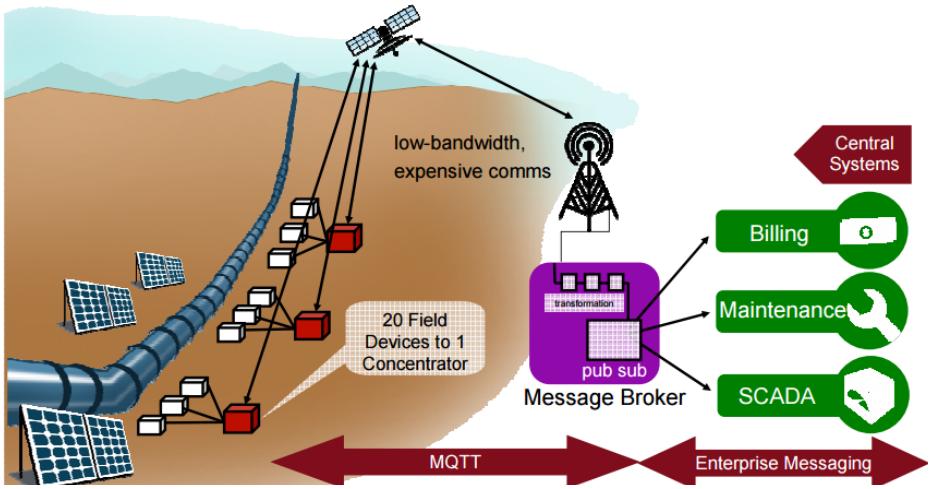


Kreps (2015) "Putting Kafka to Use"



Slack (2013-)

- **Internet of Things:**
  MQTT (Message Queuing Telemetry Transport)



Dave Locke (2012) "MQTT Connecting the Internet of Things"

A Telecommunications Typology

	Synchronous/ Asynchronous*	Unicast/Multicast	Finite/Potentially Infinite Communications
Telegraph loop	Synchronous	Broadcast	Finite
Telegraph relay	**Asynchronous**	Unicast (at each relay)	Finite
Stock Ticker	Synchronous	Broadcast	**Potentially Infinite**
Telephony (line-switched)	Synchronous	Duplex	Finite (each phone call has setup/teardown)
Radio	Synchronous	Broadcast	**Potentially Infinite**
Ethernet	Synchronous	Broadcast	Finite (fixed-size frame)
IP	**Asynchronous**	Unicast	Finite (fixed-size datagram)
RPC	Synchronous	Unicast	Finite
V Kernel	Synchronous	**Multicast**	Finite
SMTP/Email	**Async**	**Multicast** (via CC)	Finite
Message-oriented middleware	**Async**	**Multicast**	**Potentially Infinite**

* Synchrony/asynchrony refers to simultaneity of sending/receiving message and not simultaneity of sender/receiver *contact*

The Potentially Infinite in the Philosophy of Time

- McTaggart

  "The Unreality of Time" (1908):

  - A-series: includes designation of a *present moment* ('tensed')
  - B-series: events ordered by an *earlier-than* relationship ('tenseless')
    - Facts about B-series are 'fixed'

- Bergson
  - *Time and Free Will* (1899): consciousness characterized by a temporal "qualitative multiplicity", called *duration (durée)*
  - *Creative Evolution* (1907): popular knowledge about time comparable to "the contrivance of the cinematograph"

Asynchrony and the Potentially Infinite in the Philosophy of Time

- Bachelard

  The Dialectic of Duration (1936)

  - "We very soon saw that between this passing of things and the abstract passing of time there is no synchronism..."
  - "When we examined this phenomenology in its contexture... we saw that it always comprises a *duality of events and intervals*."

- G.H. Mead

  The Philosophy of the Present (1932)

  - For Mead, the past "…is expressed in irrevocability".
  - Emergent events are how we know time
  - "The social character of the universe... we find in the situation in which the novel event is in both the old order and the new which its advent heralds. *Sociality is the capacity of being several things at once*"
    - (Semiotic) implication: the more possible interpretations of a sign (e.g. as in multicast communication), *the more social the sign*

Data

(static; finite)

Codata

(dynamic; potentially infinite)

Functional programming is a good idea, but we haven't got it quite right yet. What we have been doing up to now is weak (or partial) functional programming. What we should be doing is strong (or total) functional programming - in which all computations terminate. We propose an elementary discipline of strong functional programming. A key feature of the discipline is that we introduce a type distinction between *data*, which is known to be finite, and *codata*, which is (potentially) infinite.

David A. Turner (1995) "Elementary Strong Functional Programming" in *Springer Lecture Notes in Computer Science* 1022, pp. 1-13.

Unix: the Codata Operating System (con't.)

Mahoney: To me one of the lovely features of pipes is the way it reinforces the notion of toolbox ...

McIlroy: Not only reinforces, but almost created it.

Mahoney: Well, that was my question. Was the notion of toolbox there before pipes ...?

McIlroy: No.

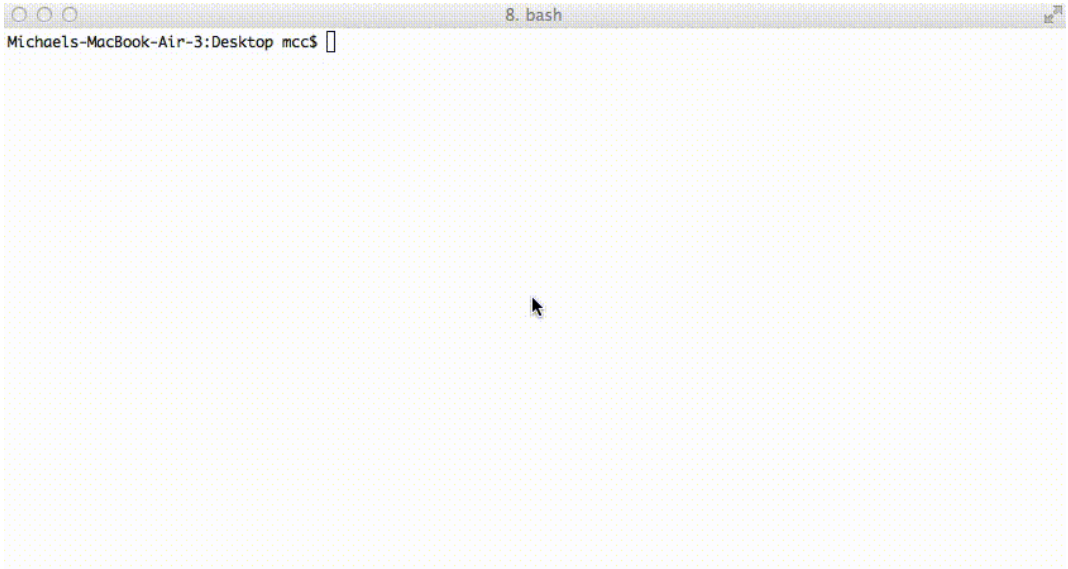
Mahoney: or did pipes create it?

McIlroy: Pipes created it.

Mahoney: Unix looked different after pipes?

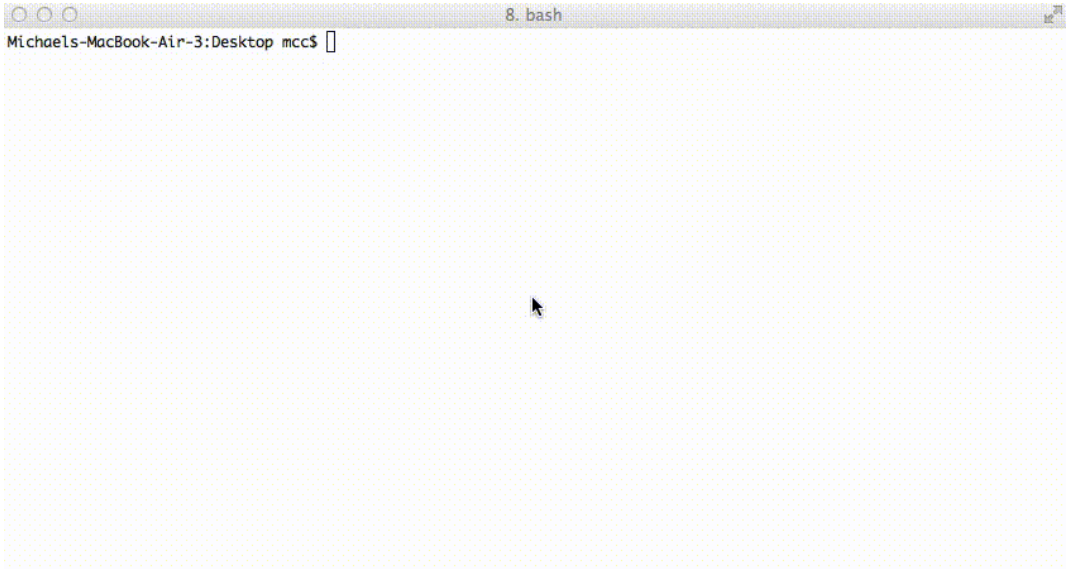
McIlroy: Yes, the philosophy that everybody started putting forth, "This is the Unix philosophy. Write programs that do one thing and do it well. Write programs to work together. Write programs that handle text streams, because that is a universal interface." All of those ideas, which add up to the tool approach, might have been there in some unformed way prior to pipes, but they really they came in afterwards.

Mahoney (1989) "Interview with M.D. McIlroy"



Unix: the Codata Operating System (con't.)

- Ritchie (1984) "A Stream Input-Output System"
  - "A stream is a full-duplex connection between a user's process and a device or pseudo-device. It consists of several linearly connected processing modules, and is analogous to a Shell pipeline, except that data flows in both directions. The modules in a stream communicate almost exclusively by passing messages to their neighbors."



Against Set Theory

- Mac Lane (1986)
  Mathematics: Form and Function
  - Attempt to build all of mathematics out of set theory leads to its own "Set-theoretic Platonism"
  - "*..all the variants of Platonism shatter on the actual practice of mathematics*."
- Hagino (1987)
  Category Theoretic Approach to Data Types
  - "..in set theory it is not easy to see either the duality between injective and surjective functions or the duality between cartesian products and disjoint sums. It is in category theory that these dualities come out clearly… [o]ur slogan is: *category theory can provide a better and more natural understanding of mathematical objects than set theory*".
- Barwise and Moss (1996)
  Vicious Circles: On the Mathematics of Non-Wellfounded Phenomena
  - "The universe of sets may or may not be a suitable tool for modeling some particular phenomena we find in the world. Some other mathematical universe may well be better in some given instance."

Databases vs. Codatabases

- Coalgebras (Fox, 2001)
  - "The usual algebraic operations all have the same form—two objects are combined to form a third. Addition, multiplication, scalar multiplication, concatenation, composition, amalgamation, fusion - these are the tools of elementary algebra, and "abstract" algebra generalizes these notions. But this ignores an entire class of operations that are just as fundamental: Factorization, partition, decomposition, scattering, fission—the operations that take a single object and break it into parts. These are the *coalgebraic* operations."

Thomas F. Fox (2001) "Everybody Knows what a Coalgebra is".

Codata in Functional Programming Research (con't.)

- Codatatypes in the ML Language (Hagino, 1989)

	datatypes	codatatypes
declaration	datatype $T =$ c_1 of $A_1 \mid \cdots \mid c_n$ of A_n	codatatype $S =$ d_1 is B_1 & $\cdots$ & d_n is B_n
primitives	constructors c_i	destructors d_i
control	case statements	merge statements
typing	$c_i : A_i \to T$	$d_i : S \to B_i$
	$\dfrac{e : T \quad e_i : A_i \to B}{\begin{array}{l}(\texttt{case } e \texttt{ of} \\ \quad c_1 \Rightarrow e_1 \\ \quad \mid \cdots \\ \quad \mid c_n \Rightarrow e_n) : B\end{array}}$	$\dfrac{e_i : B_i}{\begin{array}{l}(\texttt{merge} \\ \quad d_1 \Leftarrow e_1 \\ \quad \& \cdots \\ \quad \& d_n \Leftarrow e_n) : S\end{array}}$

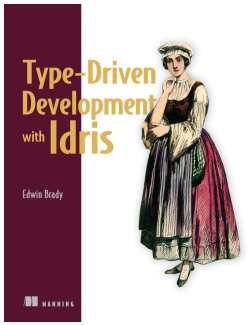
- Inductive vs. coinductive types (Vene, 2000)
  - "..inductive types (like natural numbers or lists).. provide very simple means for construction of data structures, but in order to use these values one often needs recursion.
  - "..Coinductive types (like streams, possibly-infinite lists) are *dual to inductive ones*. They come together with basic operations to destruct the values, however, their construction often involves recursion."

Hagino, "Codatatypes in ML", *Journal of Symbolic Computation* (1989) vol. 8, pp. 629-650.

Vene (2000) "Categorical Programming with Inductive and Coinductive Types".

Codata and Total Functions in Idris (Brady, 2017)

- "Idris will consider a function *total* if:
  - It has clauses that cover all possible well-typed inputs
  - All recursive calls converge on a base case"
- "The `Inf` type marks a value as *potentially* infinite, rather than guaranteeing that the value is infinite."



Edwin Brady, *Type-Driven Development in Idris* (2017)

Conclusions

- Financial services as prominent site for history and philosophy of computing
- Case for phenomenological approach to study of distributed systems
- Fundamental communication paradigms (synchronous/asynchronous, unicast/multicast) recur in layered sociotechnical contexts (optical, telegraphic, packet-switched, service-oriented)
- History and philosophy of 'Big Data' must be accompanied by its dual, the history of 'Big Codata'

Thanks!

- Charles Babbage Institute
- Computer History Museum
- Nicholson Center for British Studies, University of Chicago
- New York Stock Exchange