



# **Practicing at the Cutting Edge**

*Learning and Unlearning  
about Performance*

**Martin Thompson - @mjpt777**

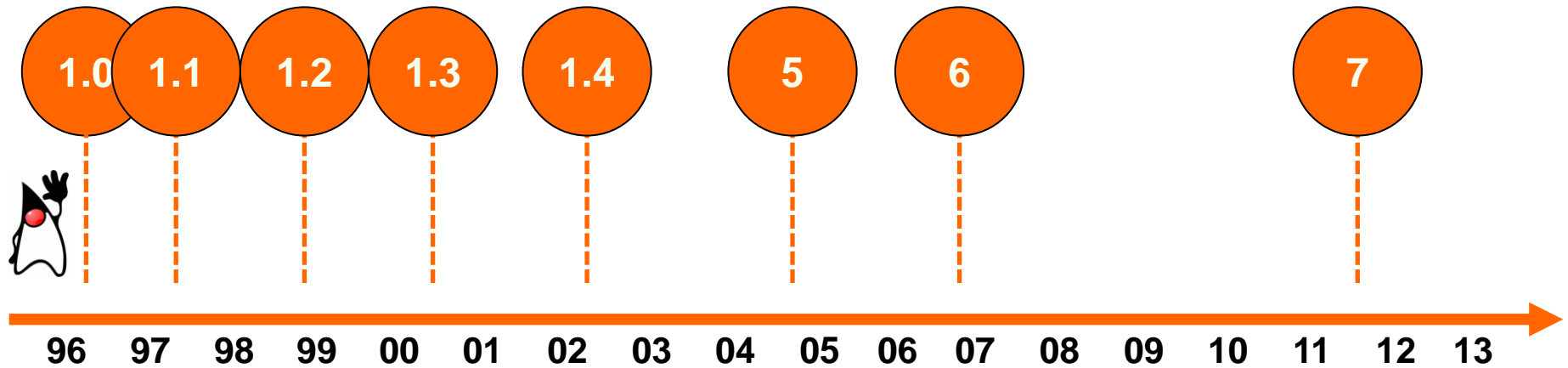
# **Learning and Unlearning**

1. Brief **History** of Java
2. Evolving **Design** approach
3. An evolving **Hardware** platform
4. Changes in **Culture**

**1.**

***Brief History of Java***

# Java Timeline



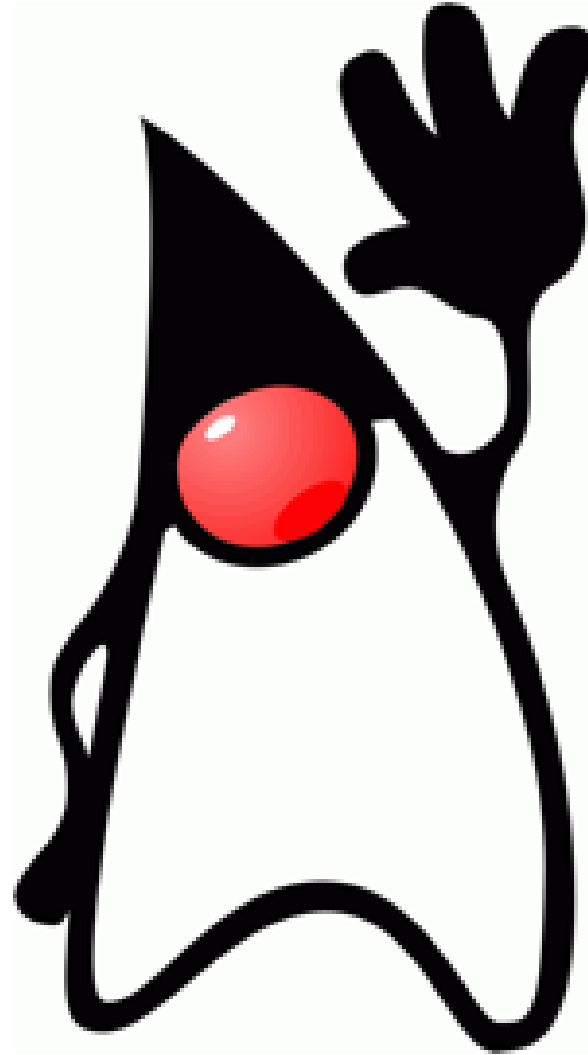
**In the beginning Java was Oak  
and Oak was slow**

**How did it all start?**











**How did Java perform then?**

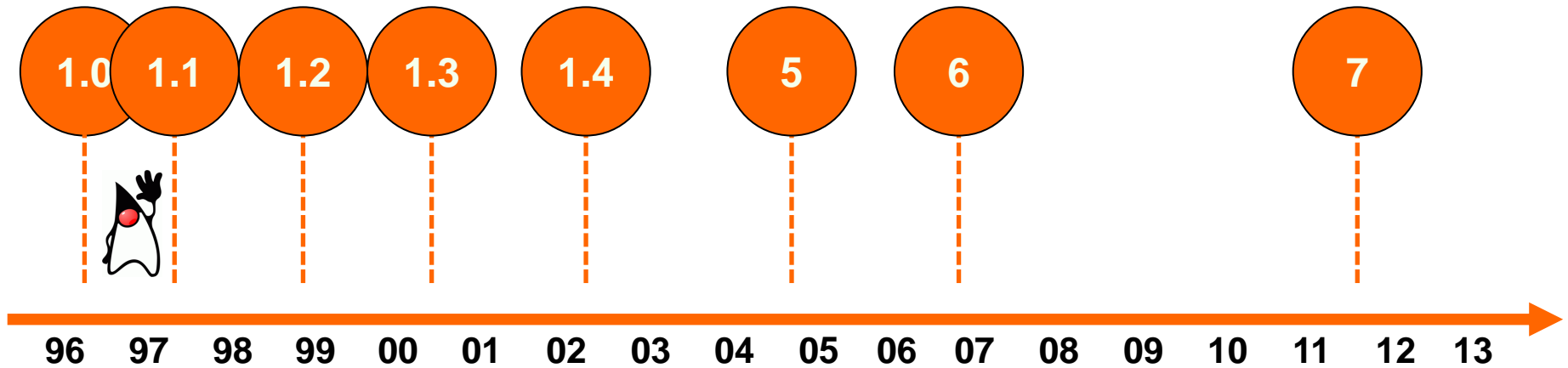






**...but everything must have a  
beginning**

# Java Timeline

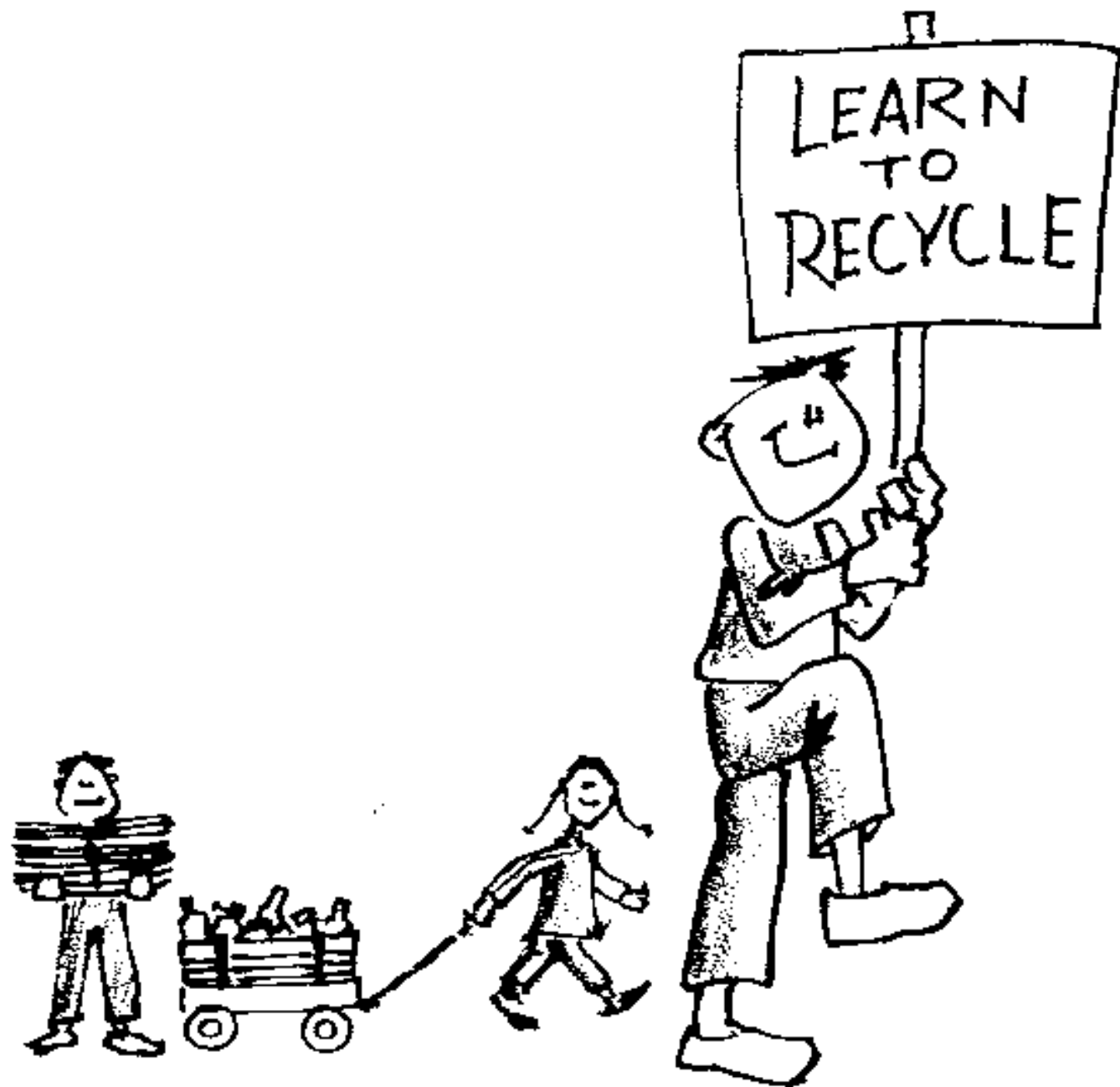


**Netscape Navigator 2.0 & IE 3.0  
with Applet support**

**Ma, I don't need to worry about deleting objects any more...**



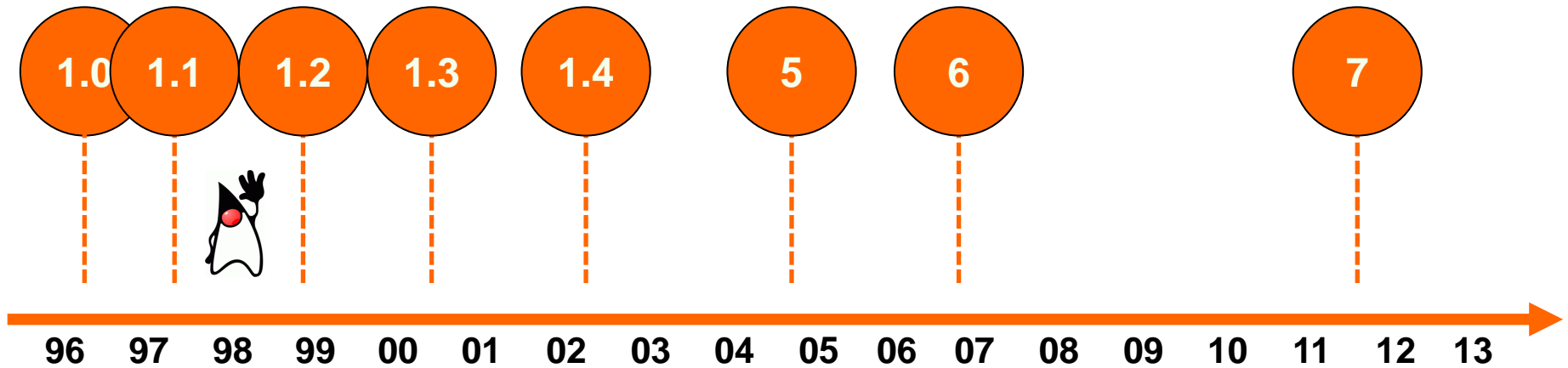




**...but we can distribute rich GUIs  
without an installer**

**We could even do multi-threaded  
apps on Windows 3.11**

# Java Timeline



## Symantec and Microsoft JIT Performance Wars



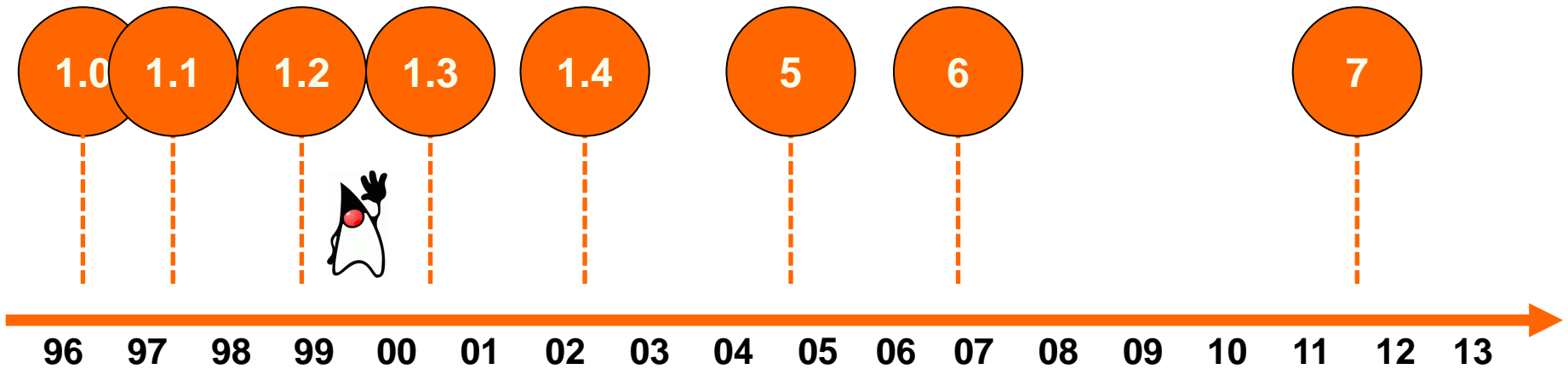


# Life with early JIT Compilers

**...cross platform GUIs are  
a real possibility**



# Java Timeline



## Generational Garbage Collection Servlets









**But then came the surprise win**

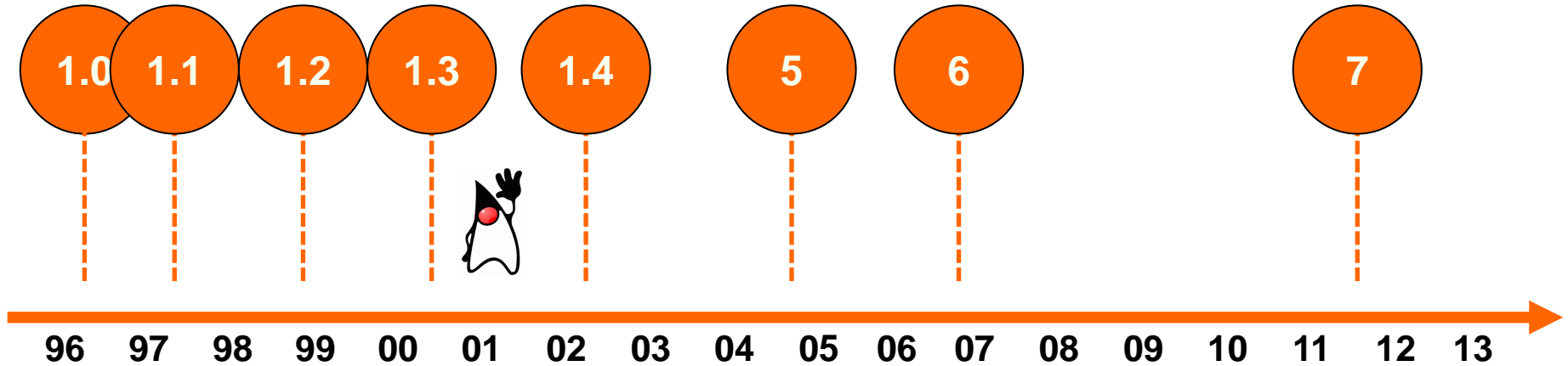
# Servlets vs CGI





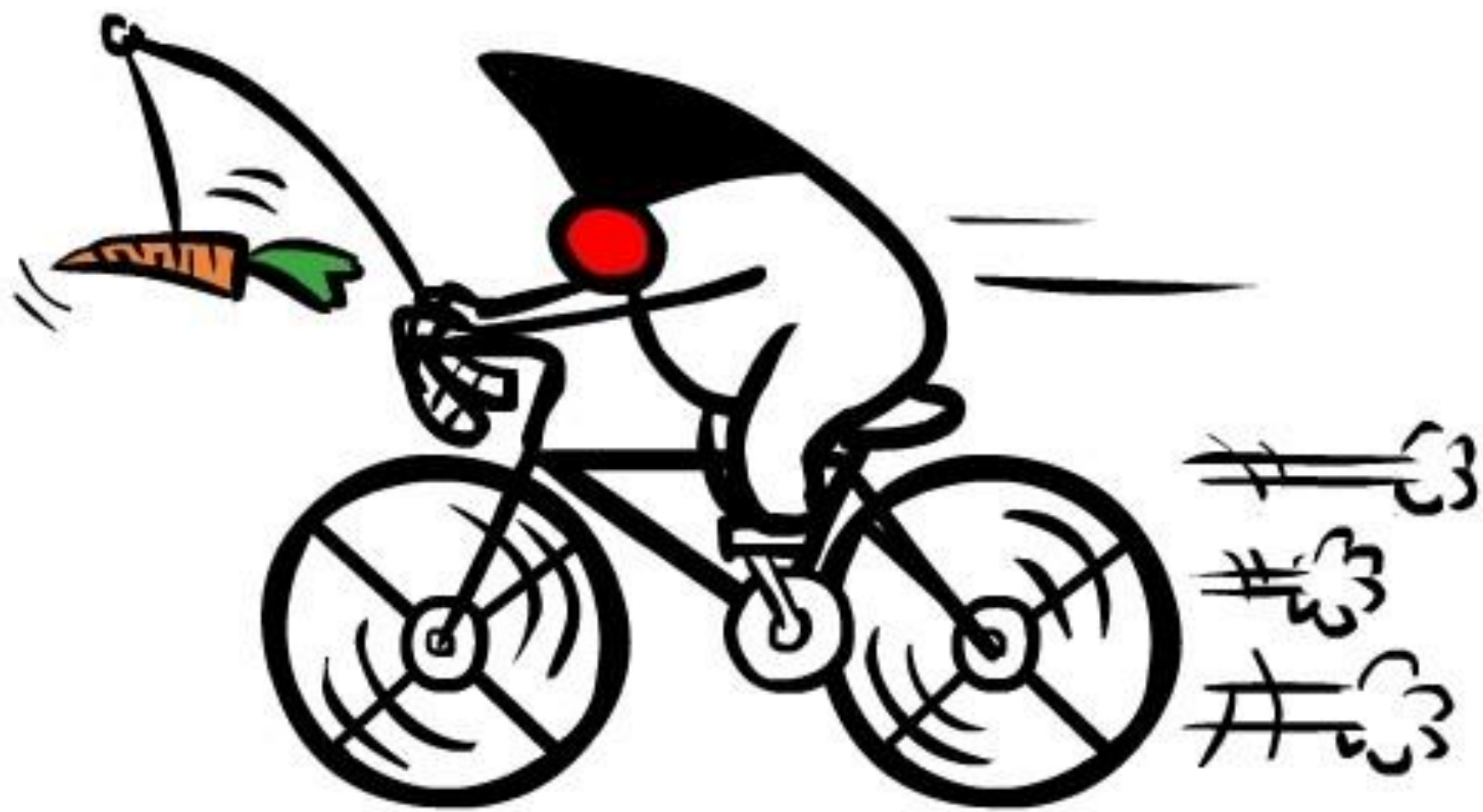
***Graph Theory!***

# Java Timeline

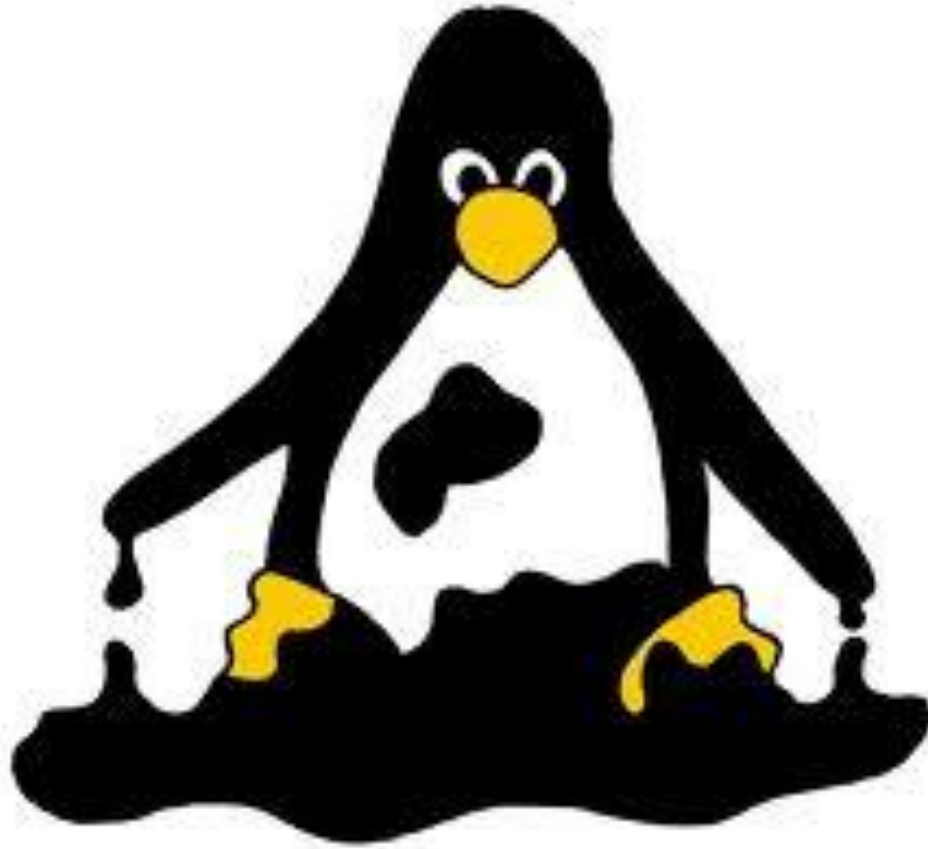


**Hotspot Compiler**  
**JRockit for x86**



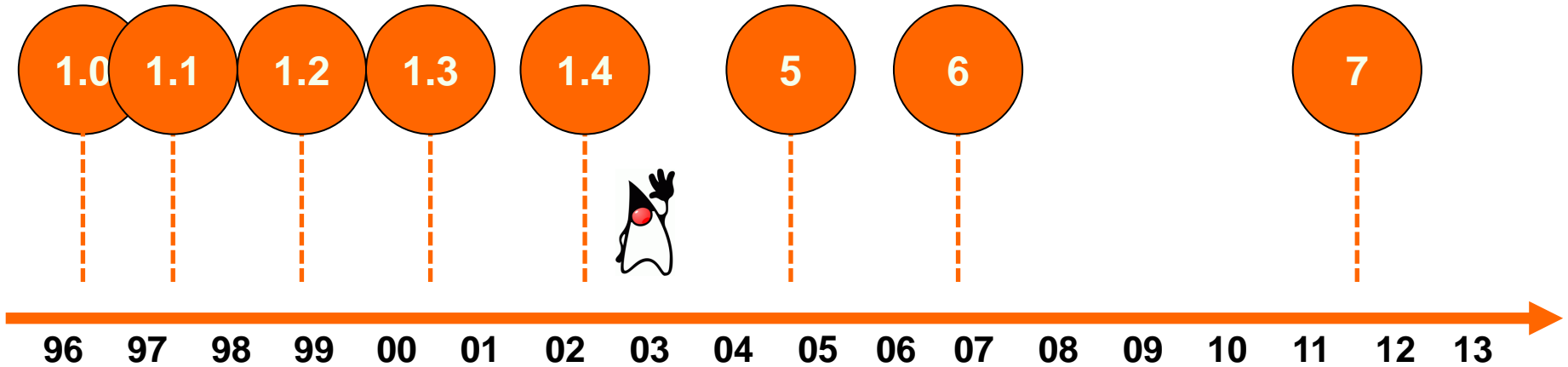


# EJB





# Java Timeline

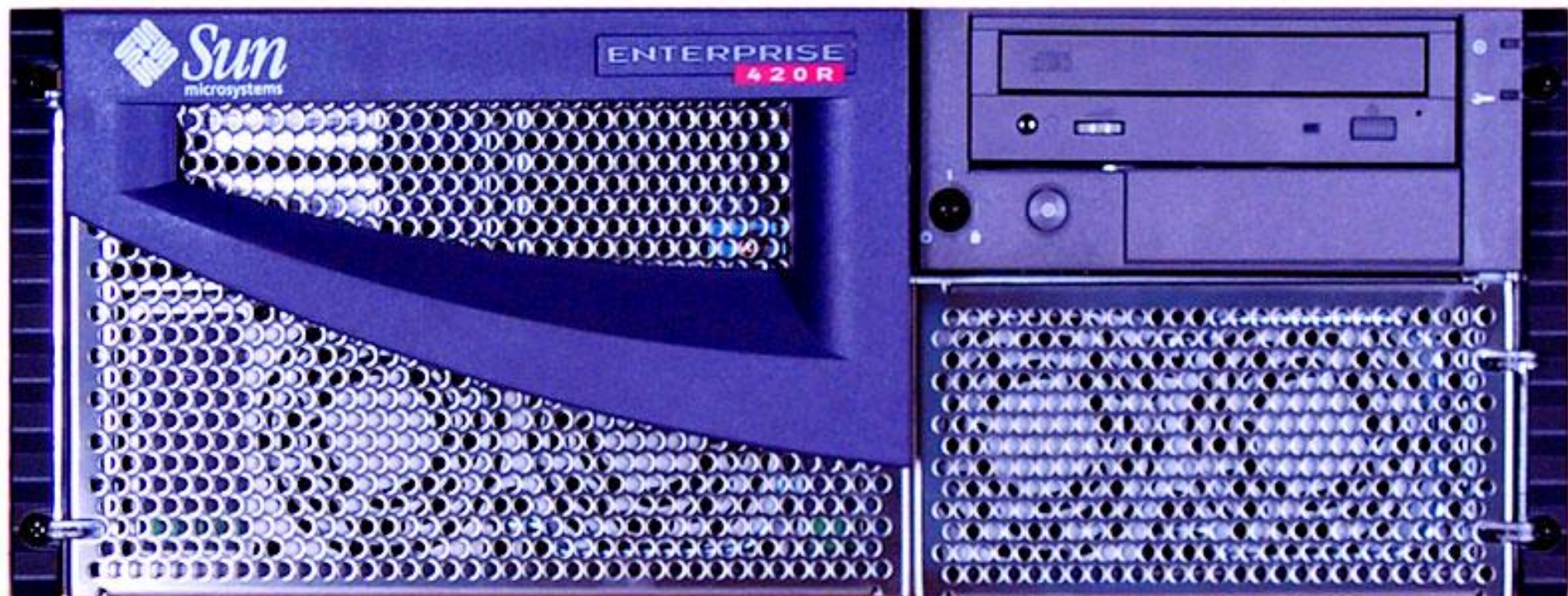


**Native Threads on Linux + epoll**  
**NIO**



# **GREATEST DOTCOM DISASTERS**



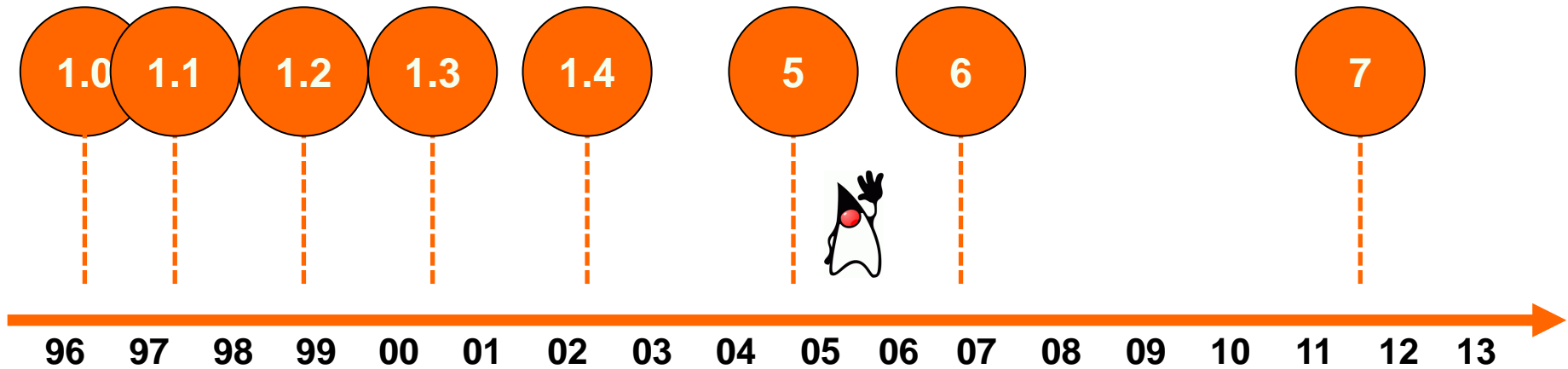




# Java Memory Model



# Java Timeline



**Java Util Concurrent + JMM**  
**Class Data Sharing**

**Herb decided to write an essay...**



***The free lunch  
is over...***



**Java finds its way into our poor  
little mobile phones**



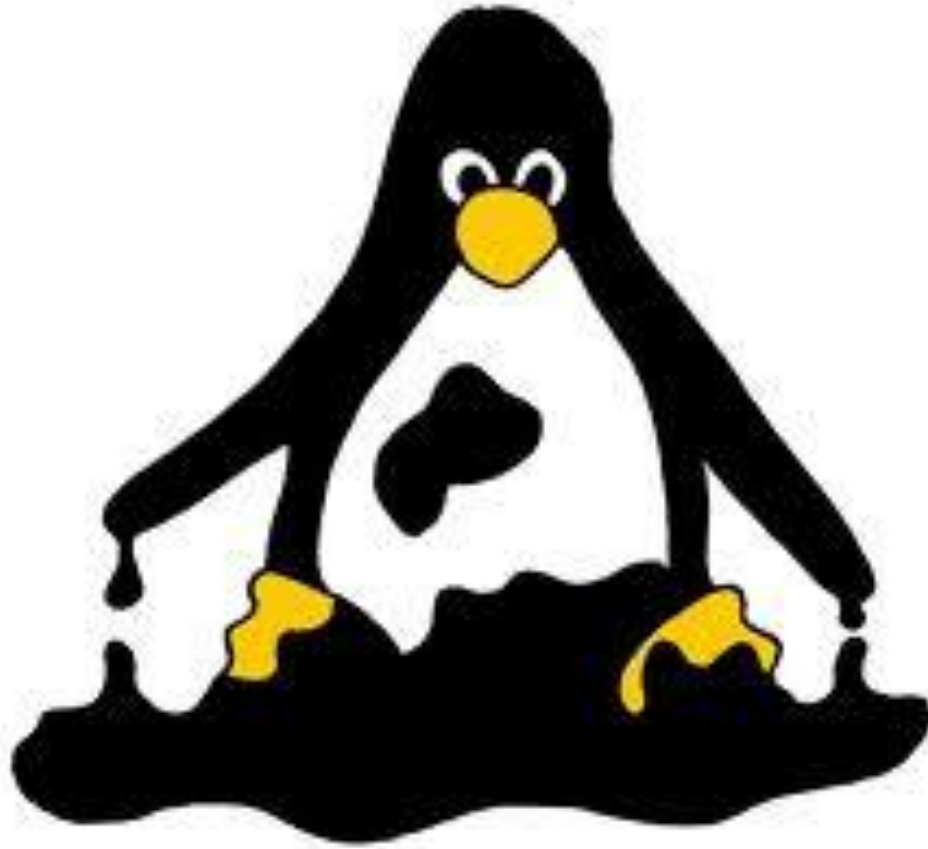
**J2ME**





**What about life on the Server?**

# EJB





Spring

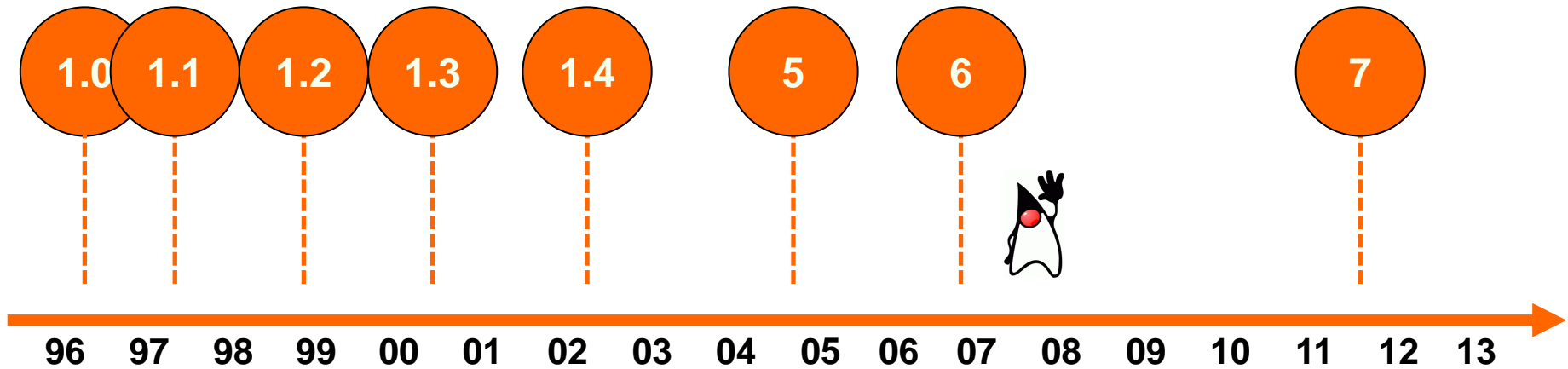
Java Application Framework

# EJB





# Java Timeline



**Escape Analysis**  
**Register Allocation**  
**Split Bytecode Verification**





**What about our hero who freed us  
from the EJB tar pit?**



Spring

Java Application Framework





**BLOATWARE**

**BLOATWARE EVERYWHERE**



HIBERNATE

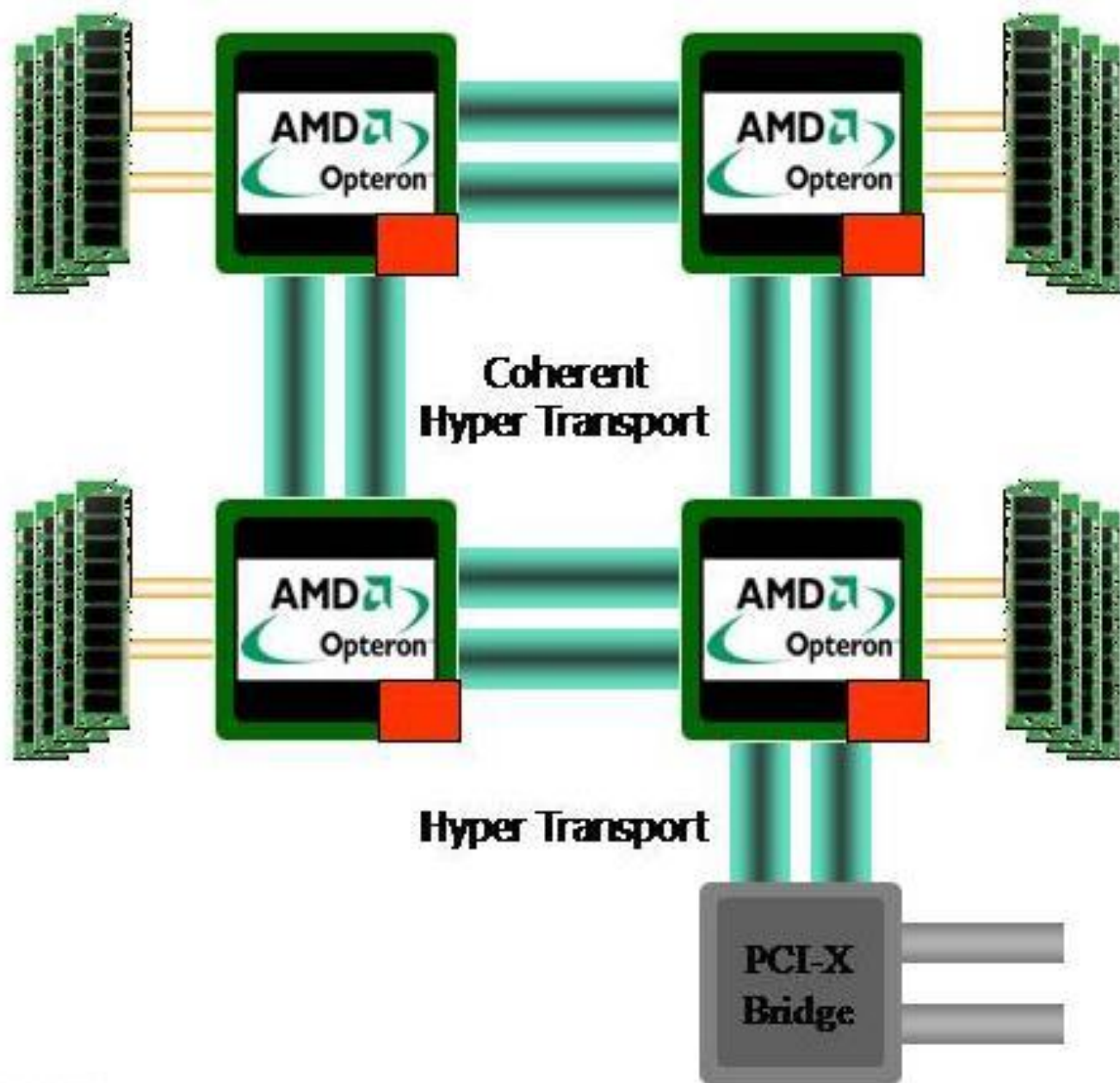




**DUMBASS!**

**SQL too difficult?**

**The world of hardware  
undergoes big changes!**





# High Frequency Trading



**Milliseconds Matter...**

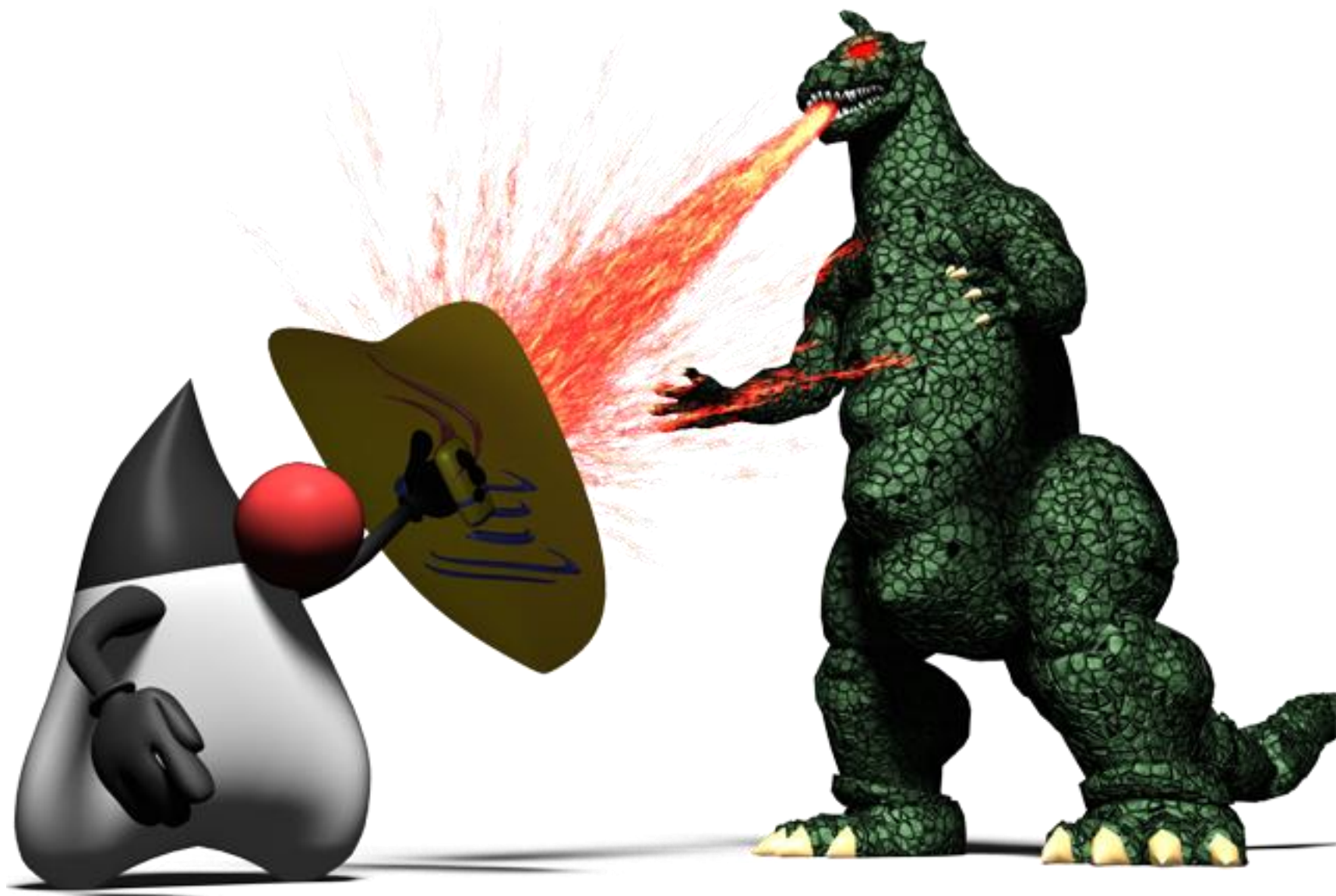
**...Microseconds  
Matter...**

**...right from market  
open...**

**...and especially after a  
quiet period...**

**...it is like being at War**









# Memory Cliff



# SHM, Locks, and Multi-core

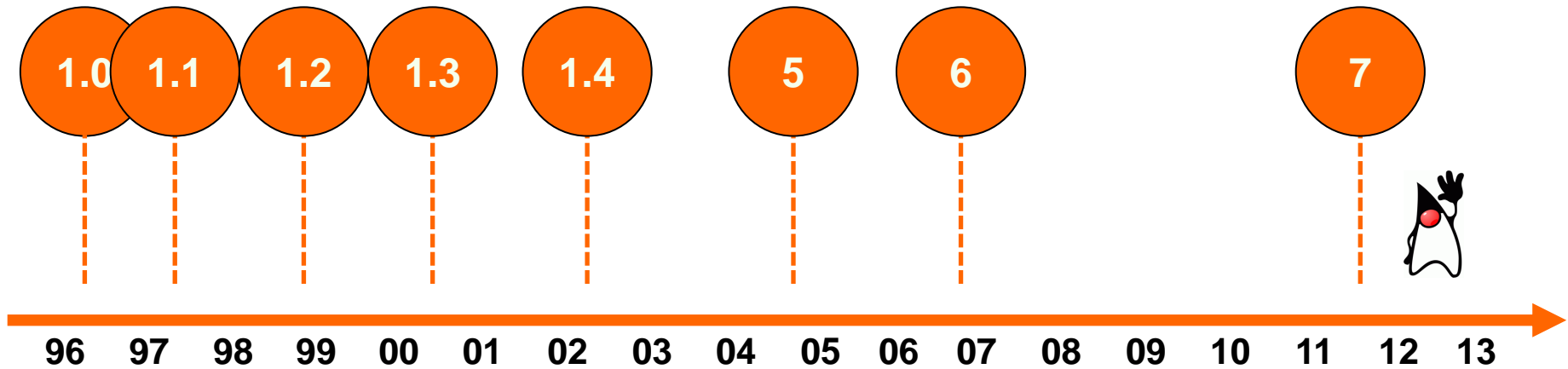




**Safepoints!!!**



# Java Timeline



**Tiered Compilation**  
**G1 Garbage Collector**  
**Compressed Pointers**



**What does the near future hold?**

JDK8



**sun.misc.Unsafe**





**WTF?!?!**

JDI 3



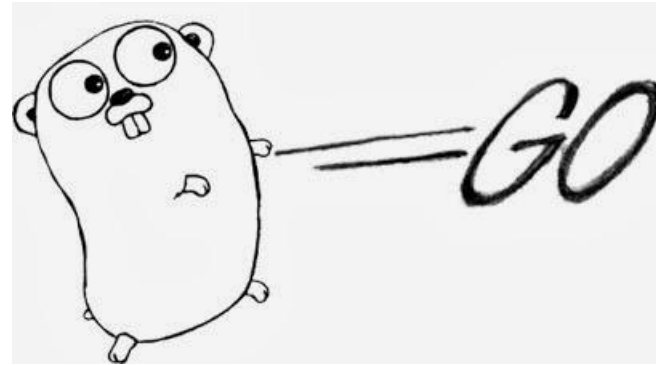
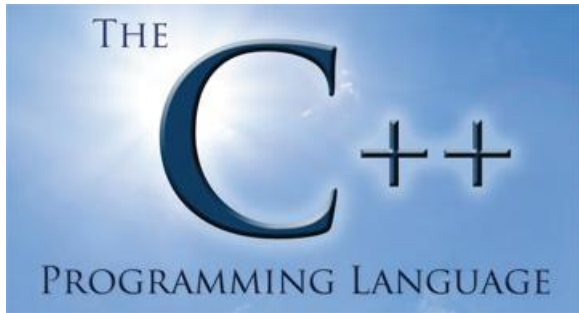
**What is everyone excited about?**



λ



**What about other languages?**



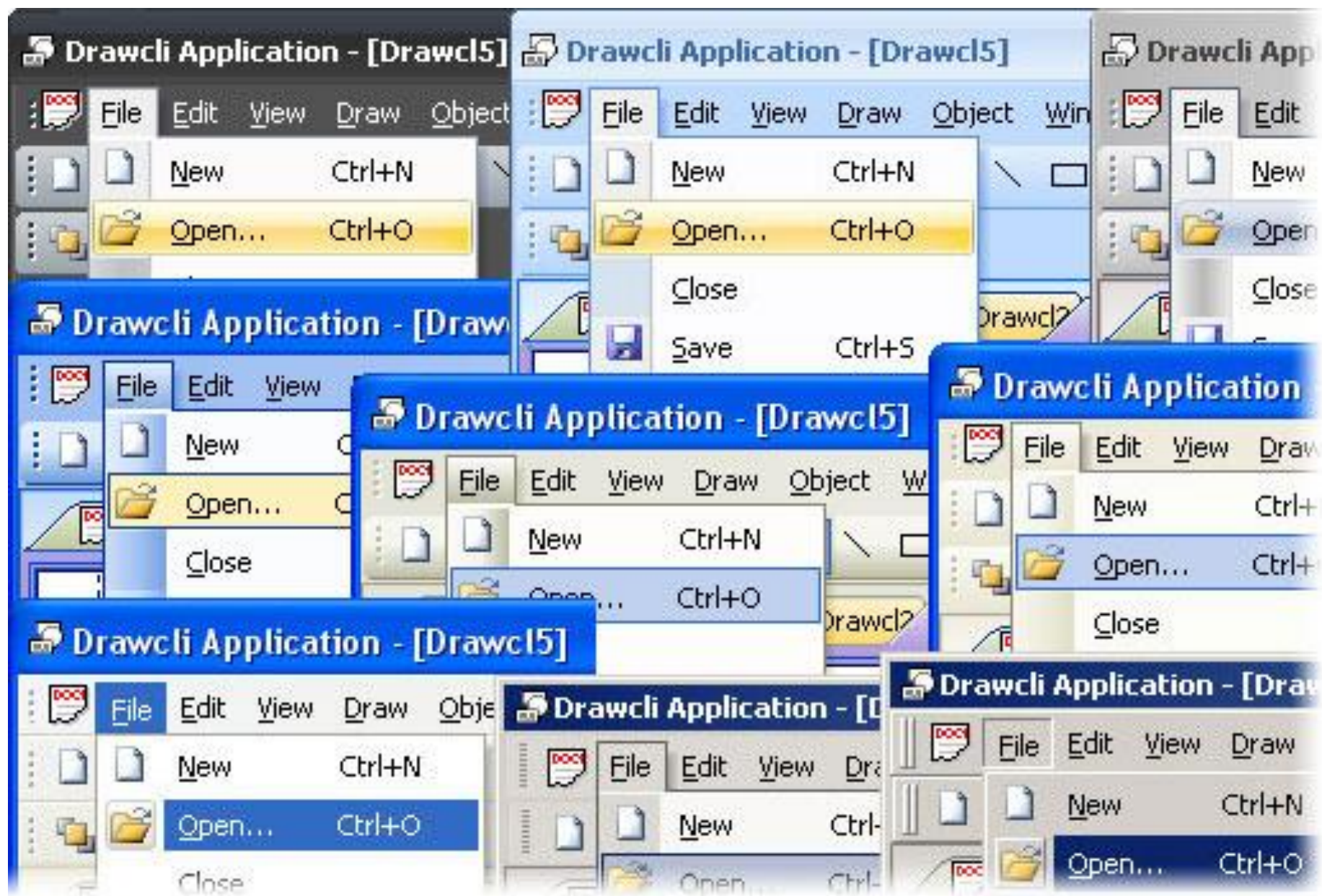
**Now on with the story...**

**2.**




***Evolving Design Approach***

# GUI Era



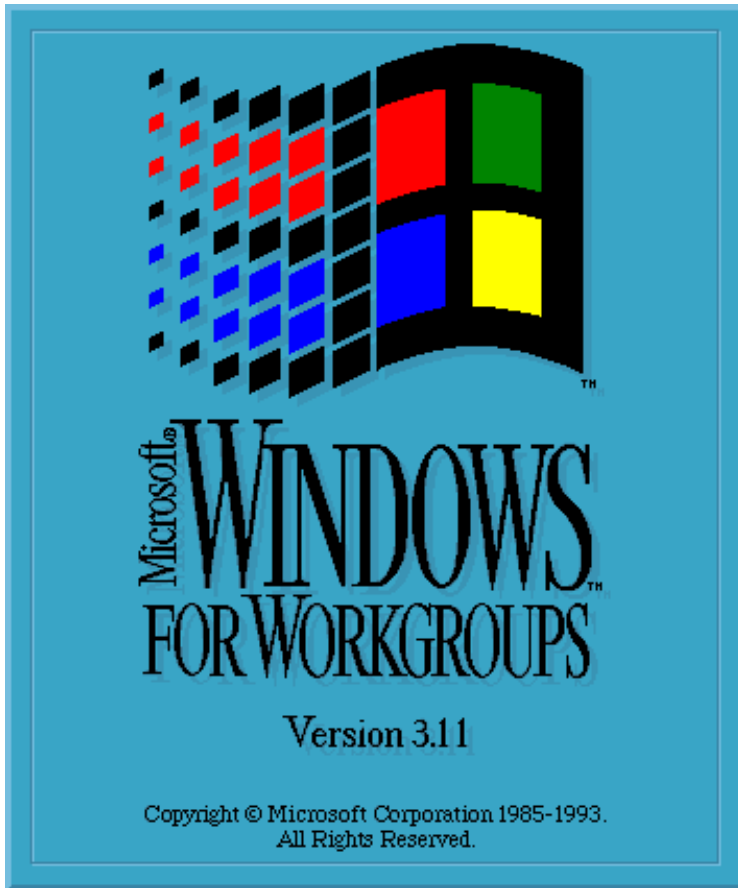


Children of the DemoTop Widget ".browseTop"

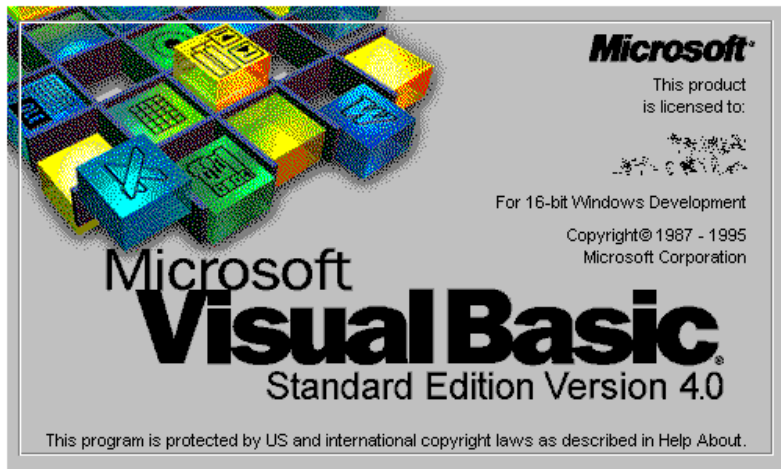
Path Name	Class	X	Y	Width	Height	Mapped	Viewable	Manager
 .browseTop.tf	Frame	0	0	533	192	yes	yes	pack
 .browseTop.menu	Menu	0	0	1	1	no	no	wm
 .browseTop.bf	Frame	0	192	533	48	yes	yes	pack

Refresh Parent Close

# Battle for the Desktop



# Battle for the Desktop



**What was the biggest  
factor in performance?**

**People**



**Comms design is  
significant to GUIs**

**Users love responsive interfaces**

# Modelling



O Object

O Oriented

P Programming

# “Big” Data Era

**What do the following  
companies have in common?**



**CONRAD**

**Office  
DEPOT**

**HILTI**

**ABB**

**RS Components**

**Huge product catalogues**

**Logistics**

**Partners**

**Catalogues**

**Sales**

**PCM**

**Websites**

**Warehousing**

**e-Procurement**

**Manufacturing**

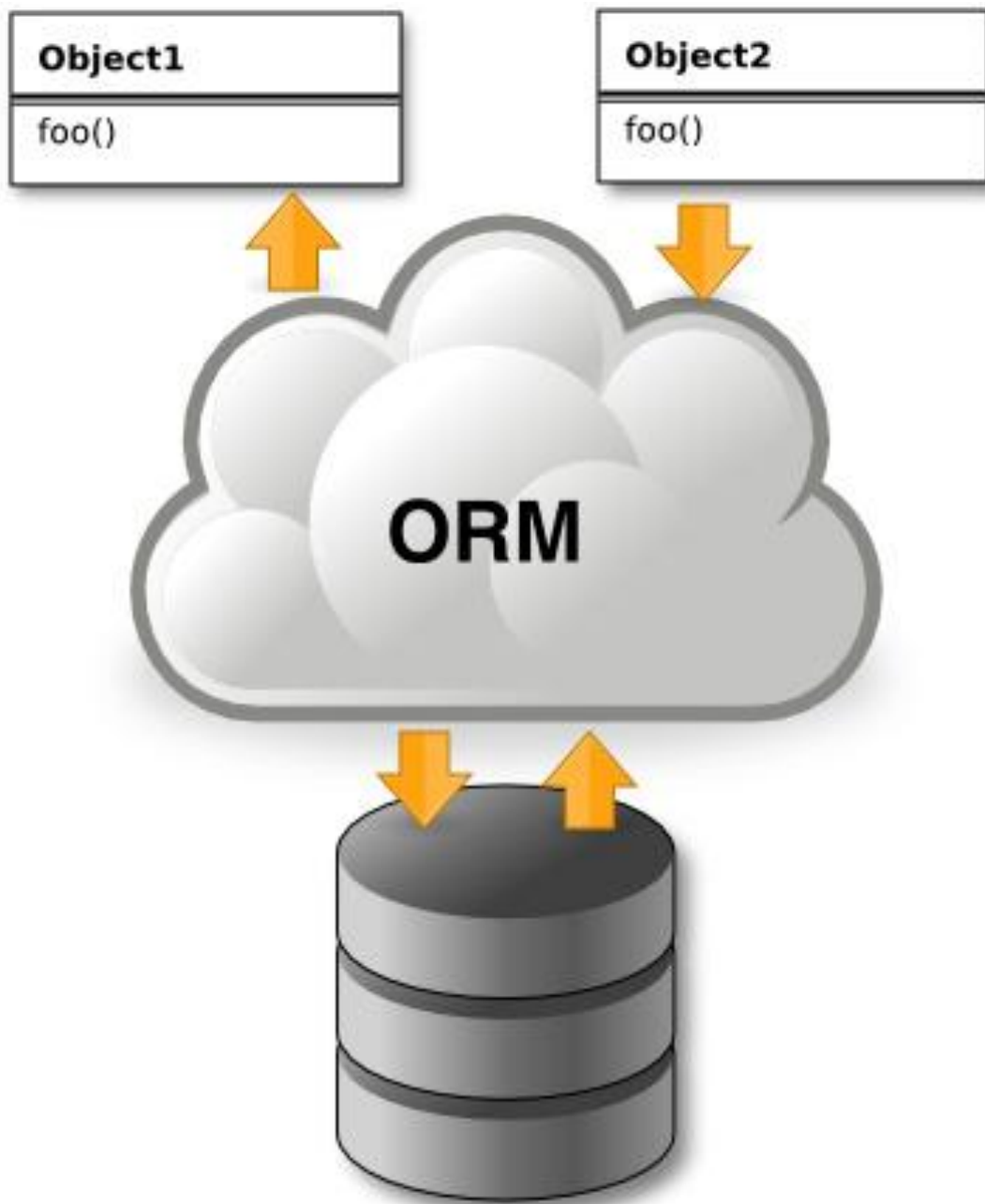
# Parsing

# **Data Modelling & Representation**

# Indexing







<b>Object1</b>
foo()

<b>Object2</b>
foo()



**FAIL**



**Set Theory Rocks!**

**Databases are very powerful!**

**Immutable Data Rocks!**



**JDBC Drivers Suck!!!**

# **Stream Processing Rocks!**

```
public void characters(char[] ch,  
                      int start,  
                      int length)  
    throws SAXException
```

```
public void characters(char[] ch,  
                      int start,  
                      int length)  
    throws SAXException
```

```
public void startElement(String uri,  
                        String localName,  
                        String qName,  
                        Attributes atts)  
    throws SAXException
```

```
public String[] split(String regex)
```

```
public String[] split(String regex)
```

```
public Iterable<String> split(String regex)
```

```
public String[] split(String regex)
```

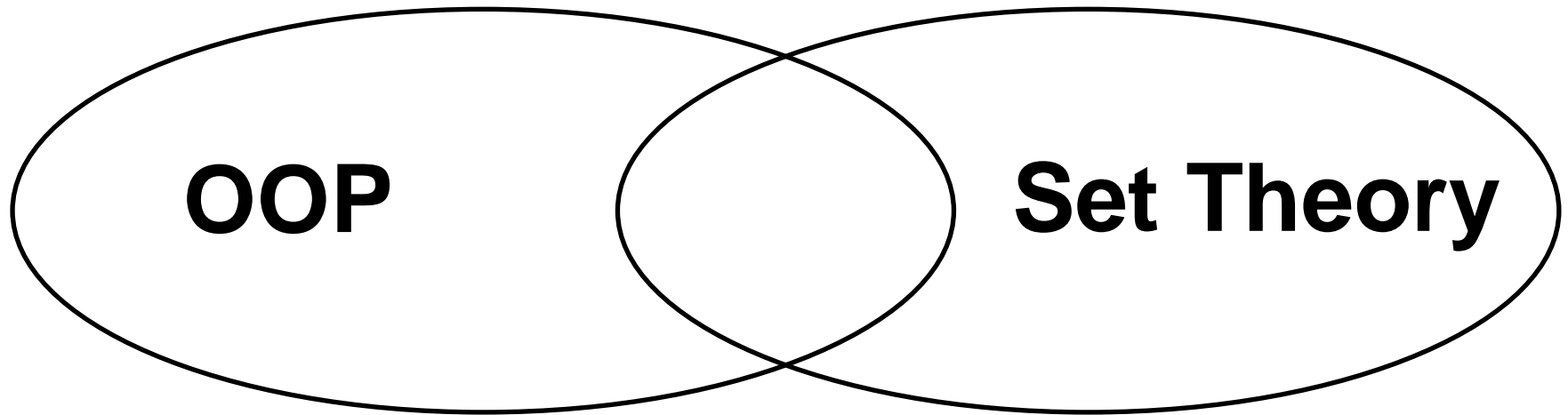
```
public Iterable<String> split(String regex)
```

```
public void split(String regex,  
                  Collection<String> dst)
```



# Modelling

# Design Paradigms



# Transaction Processing Era

 **betfair**  
sports casino poker

**LMAX**<sup>TM</sup>  
E X C H A N G E



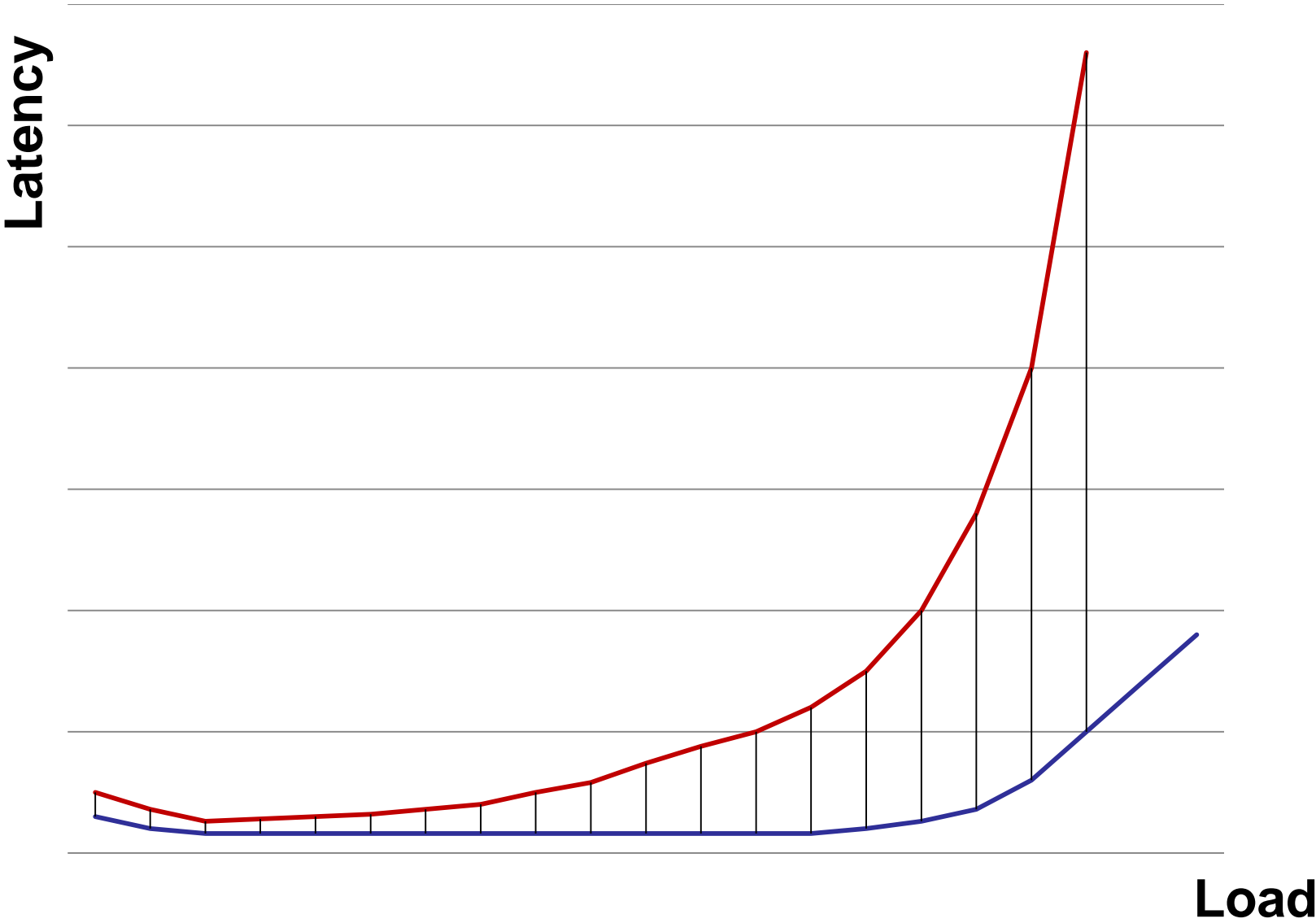
**Synchronous designs  
are seriously limited**

# **Staged Event Driven Architecture**

# **The importance of Latency**



# Latency given Throughput



# **The issues with Concurrency**

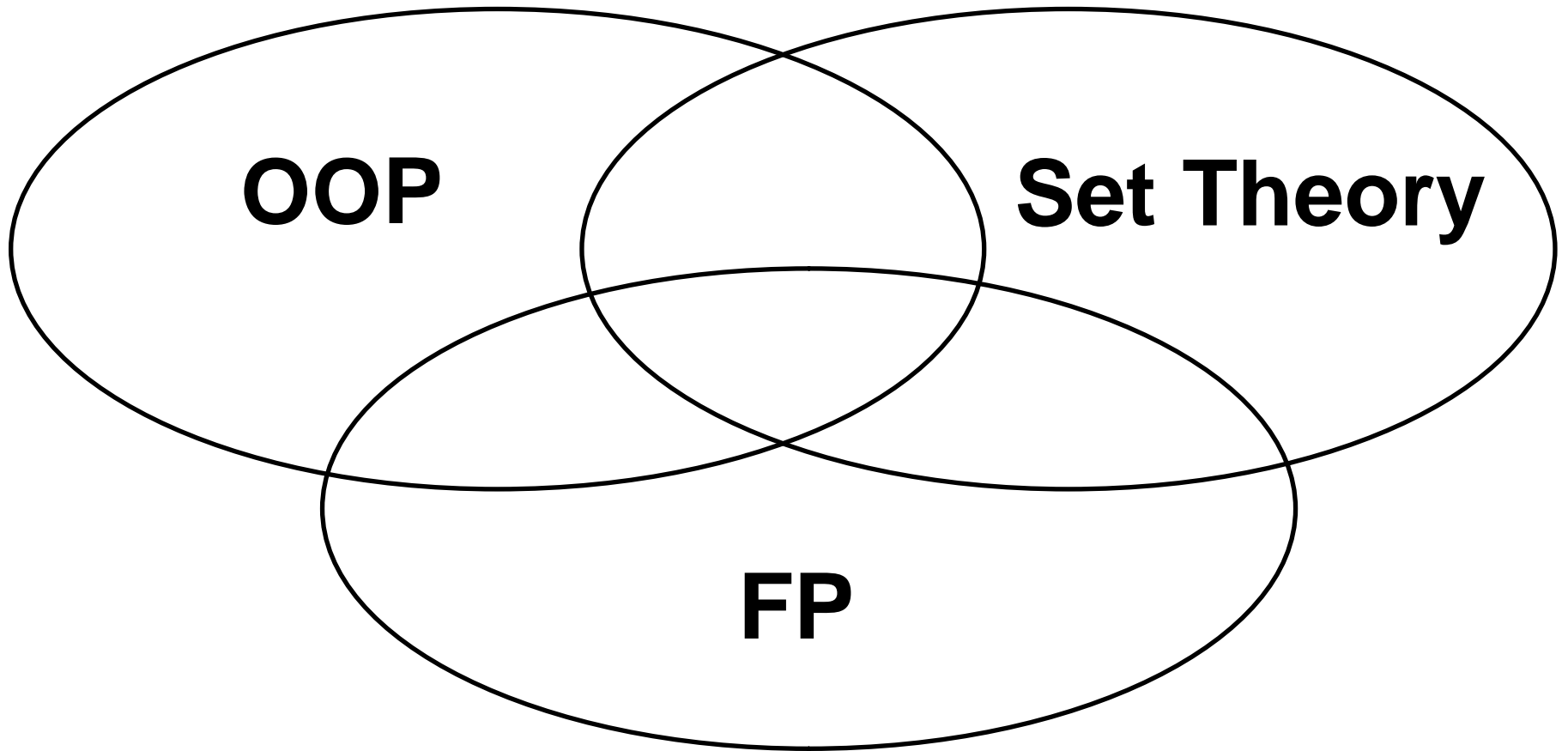
# **The Disruptor**

# **The value of Determinism**

# **The value of pure Functions**

# Modelling

# Design Paradigms



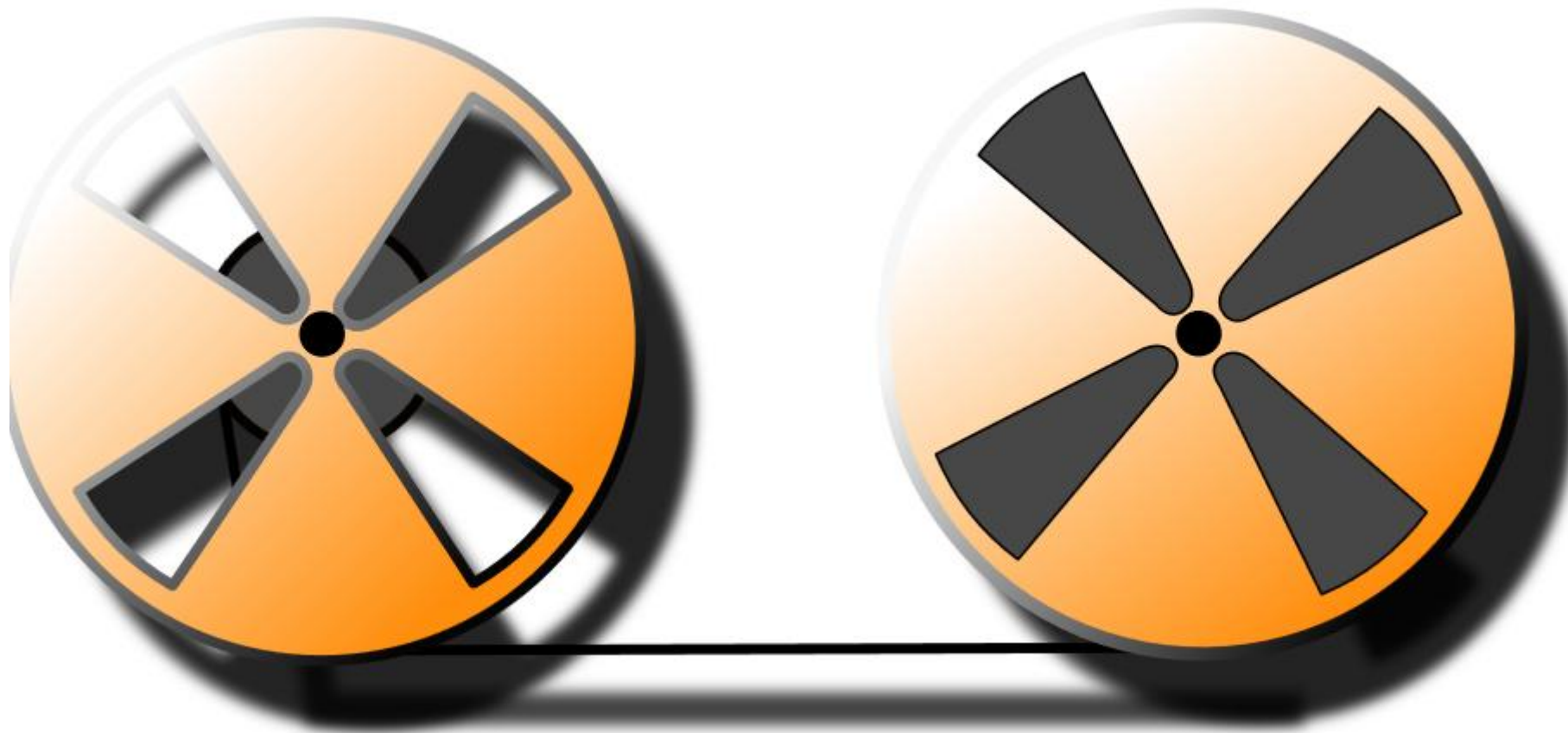


**3.**

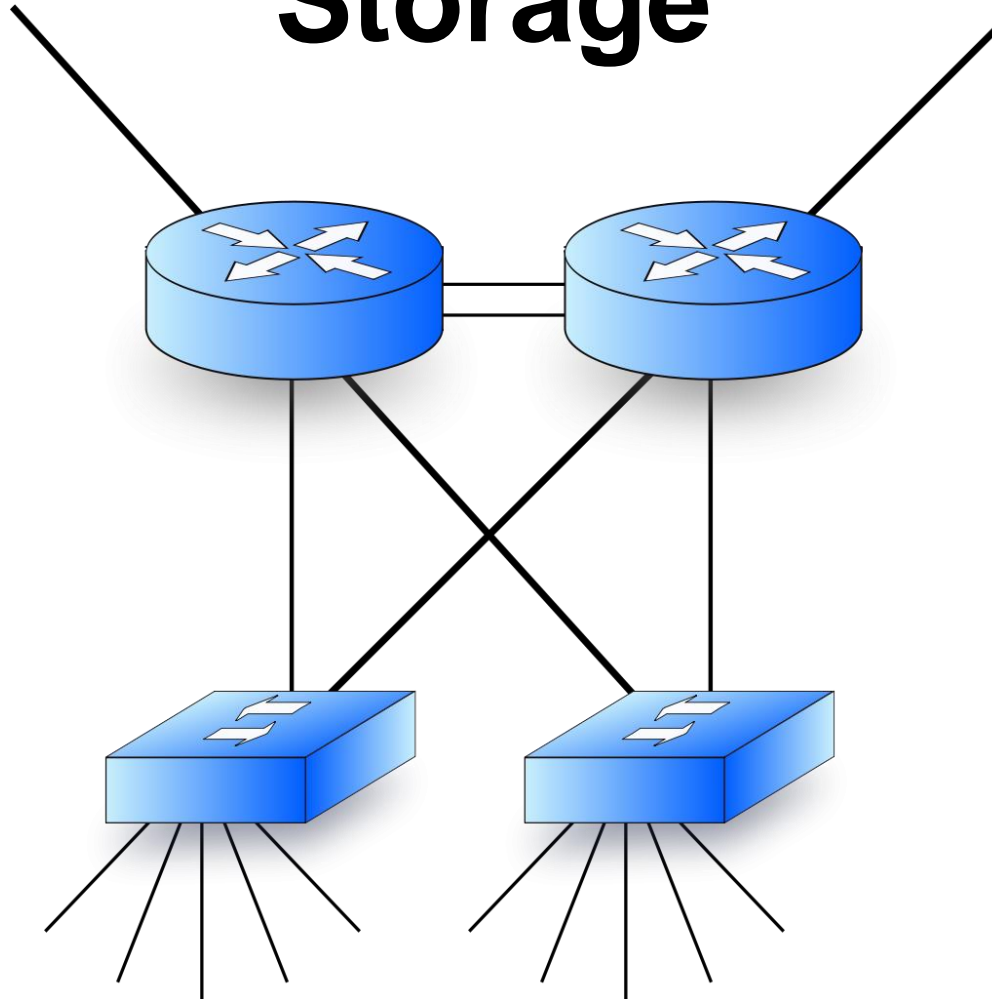
***Evolving Hardware  
Platform***

# Mechanical Sympathy

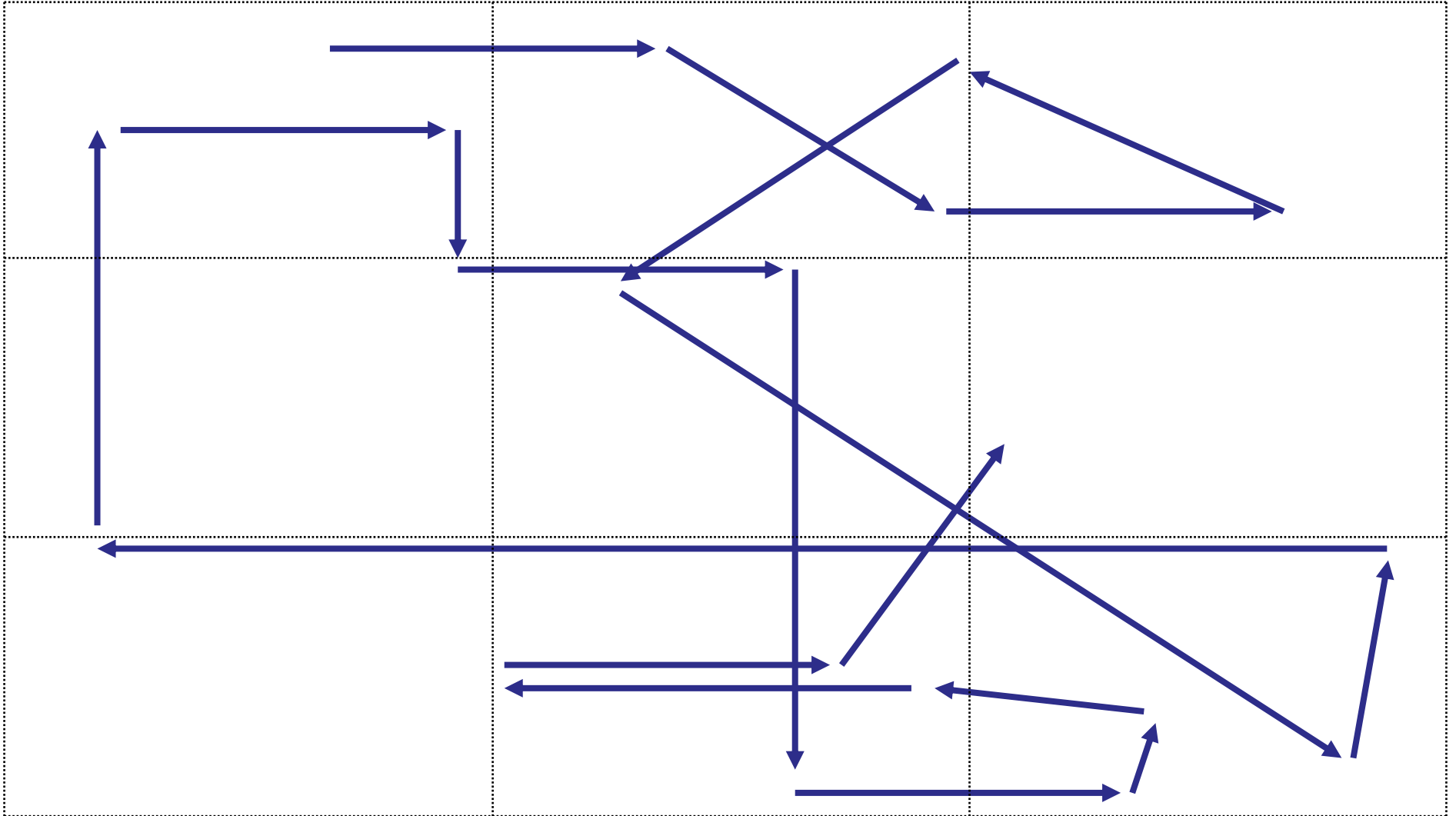
# All Storage is Tape



# Networks are faster than Storage

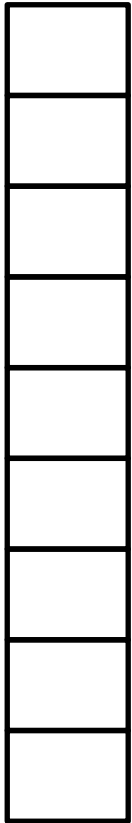


# Memory Access Patterns Matter

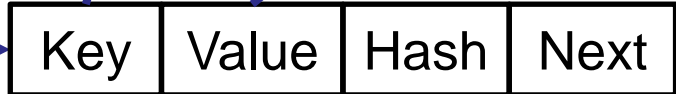
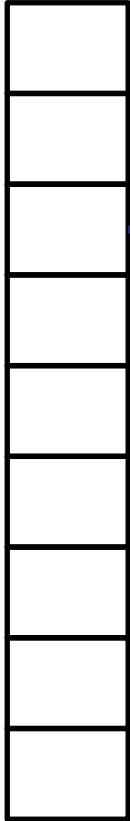


**What if you could build a HashMap  
with reified generics  
and arrays of structures?**

Buckets

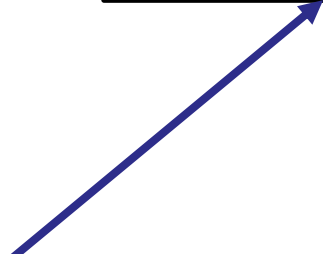


Buckets

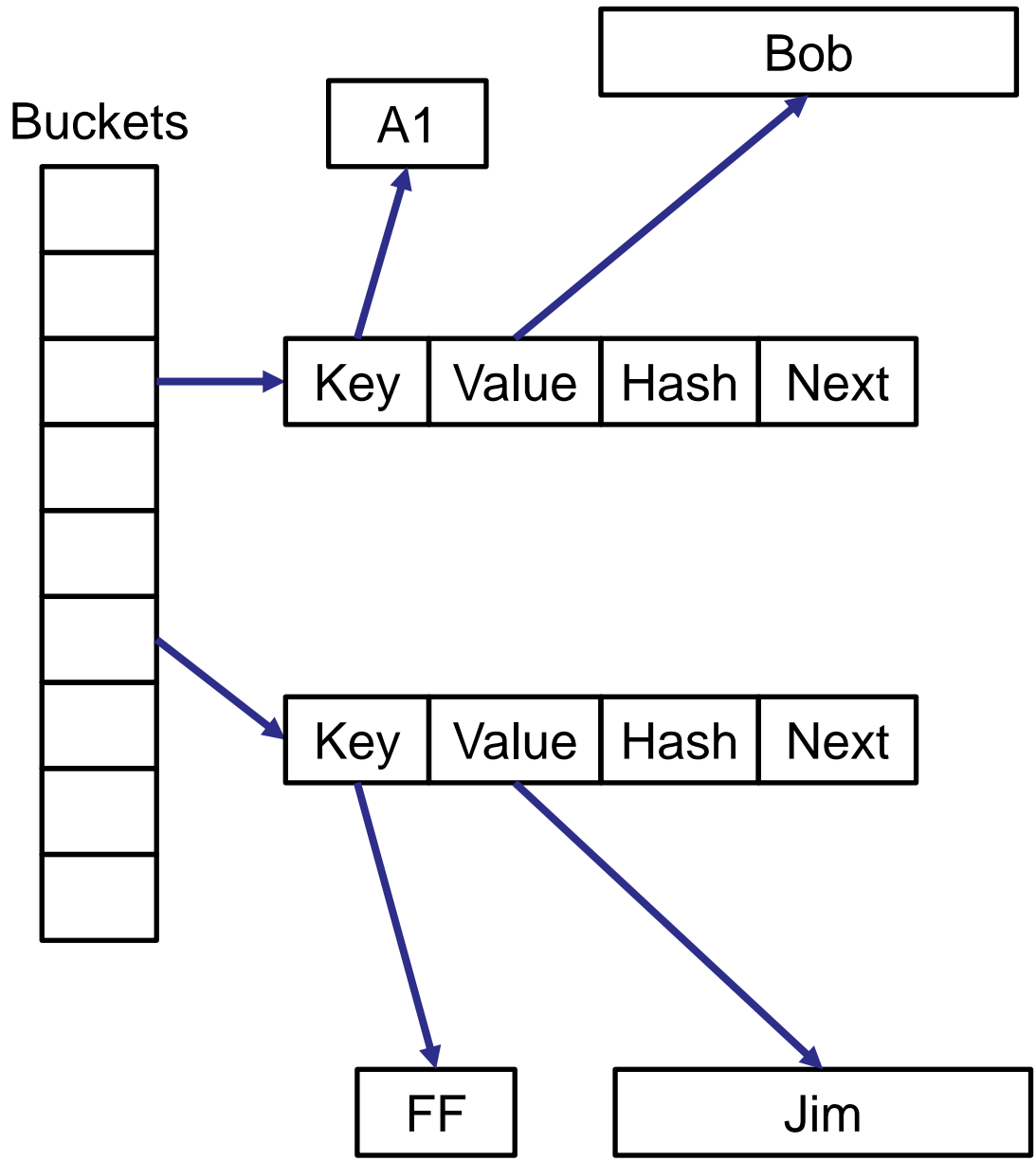


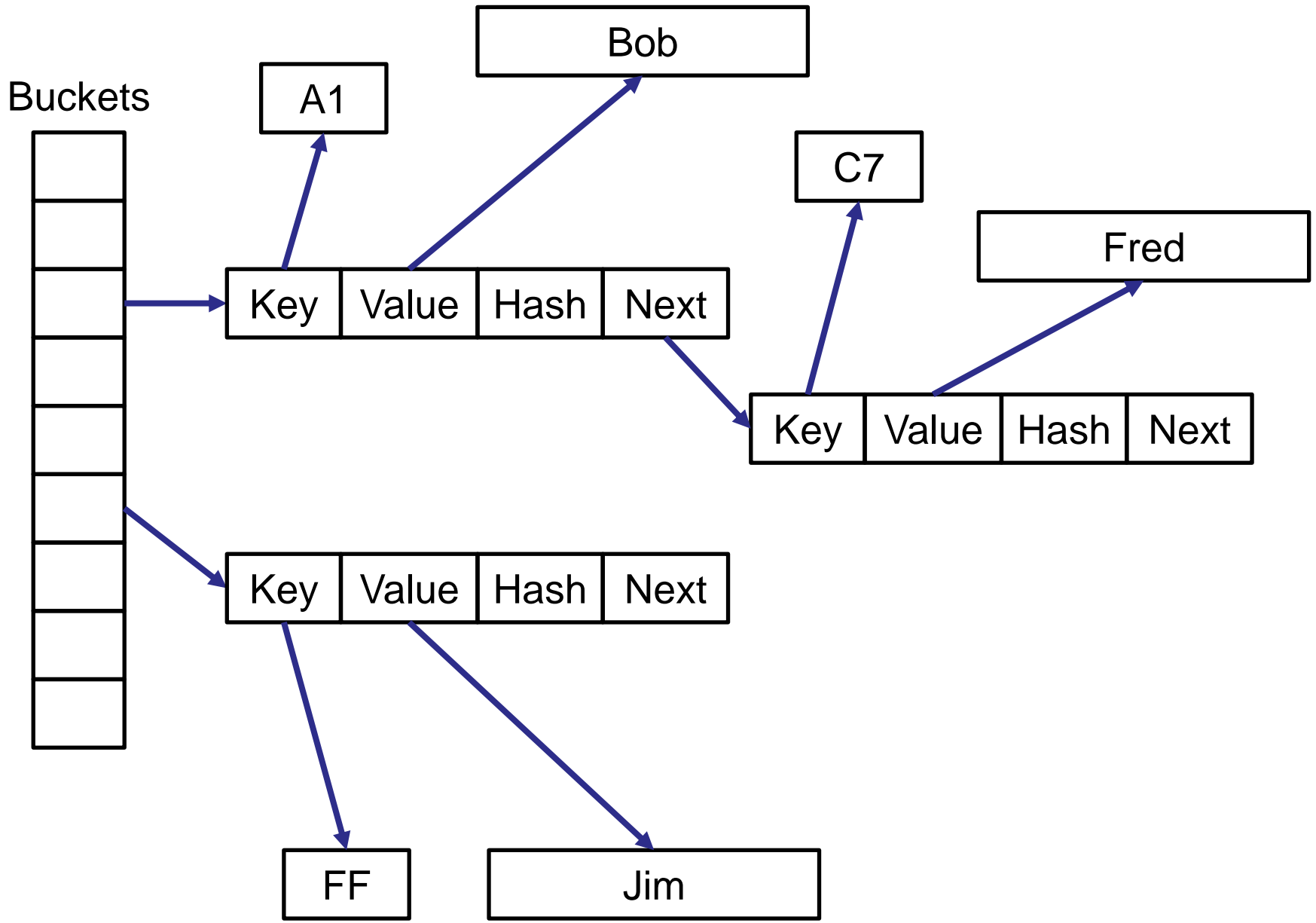
A1

Bob













Buckets

1
0

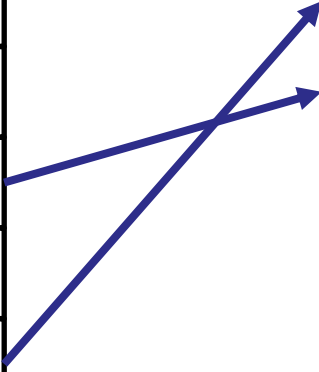
Key

Value

Hash

Next

Key	Value	Hash	Next
A1	Bob	4	-1
C7	Fred	2	-1



Buckets

1
0

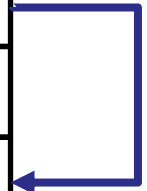
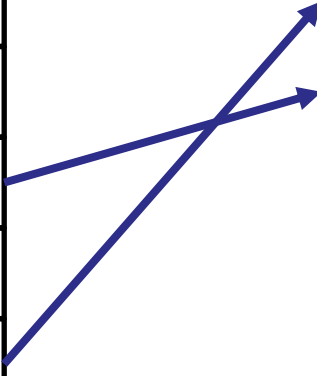
Key

Value

Hash

Next

Key	Value	Hash	Next
A1	Bob	4	2
C7	Fred	2	-1
FF	Jim	4	-1



**.Net Dictionary is 10X faster  
than HashMap for 2GB of data**

# **Value Types (Arrays 2.0<sup>64</sup>)**

**ObjectLayout**

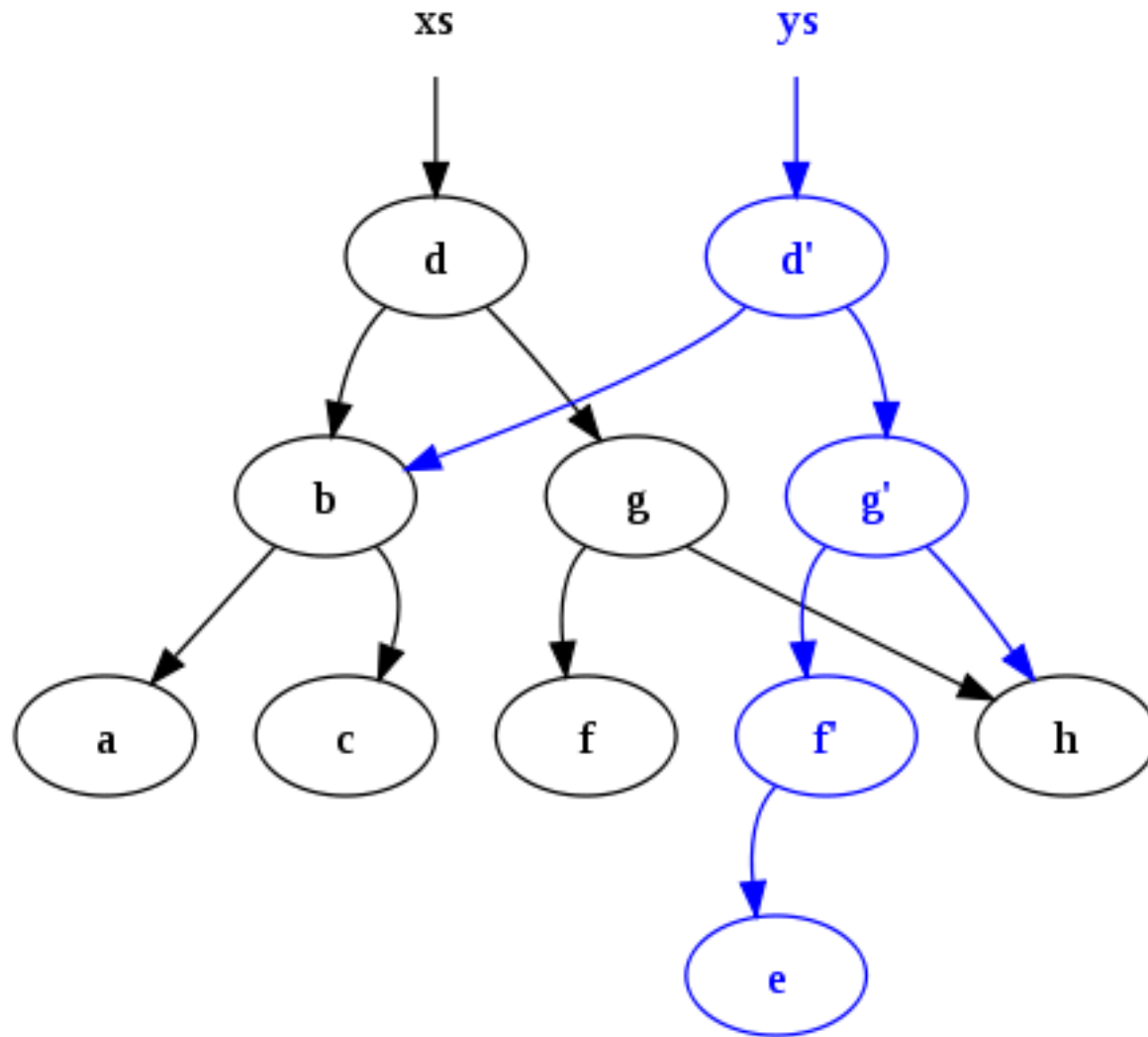
**PackedObjects**

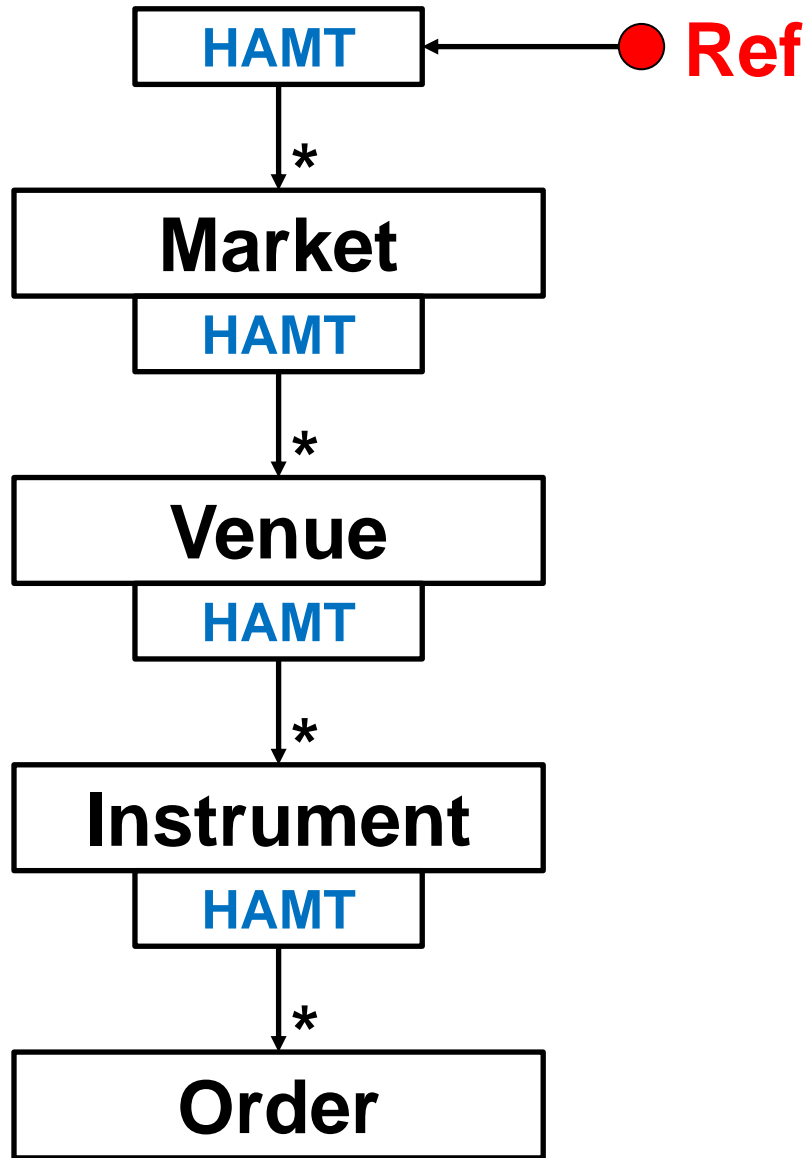


***“Functional Programming will solve  
the multi-core problem”***

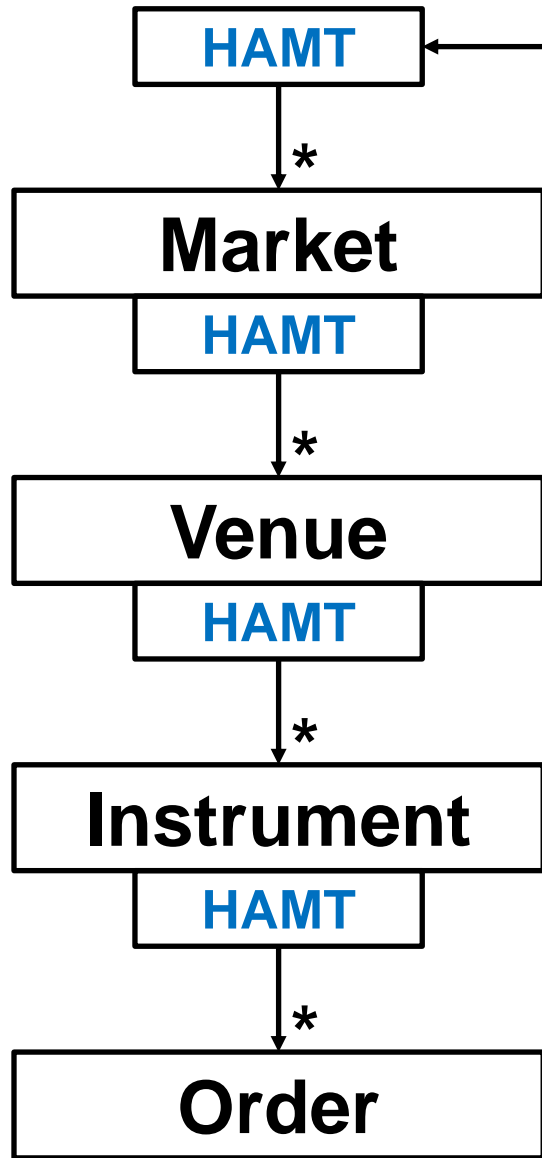
**- Functional Fanboi**

# Persistent Data Structures



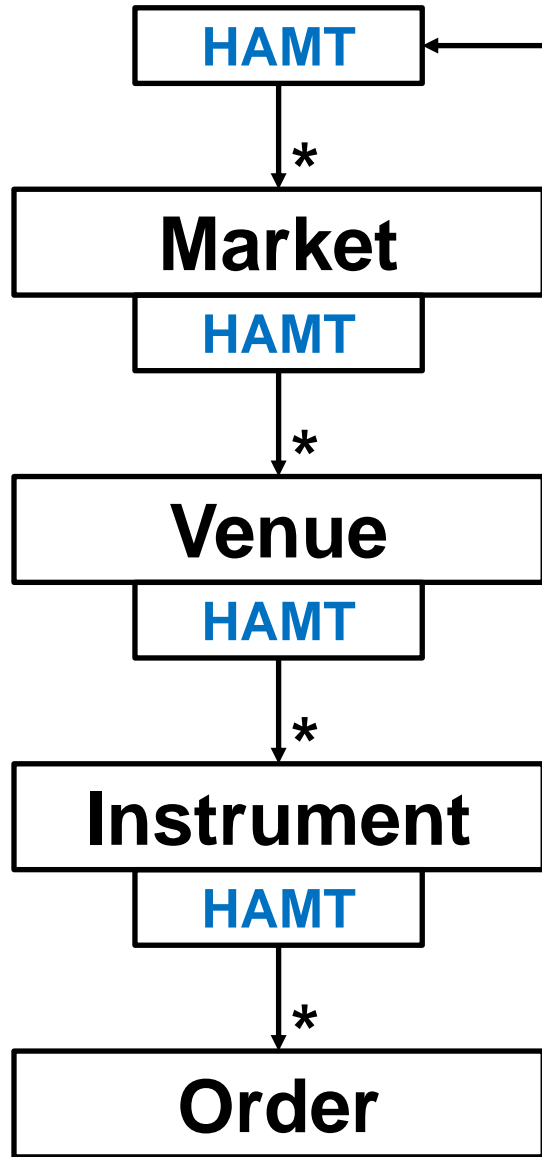


<< CAS Failure? >>

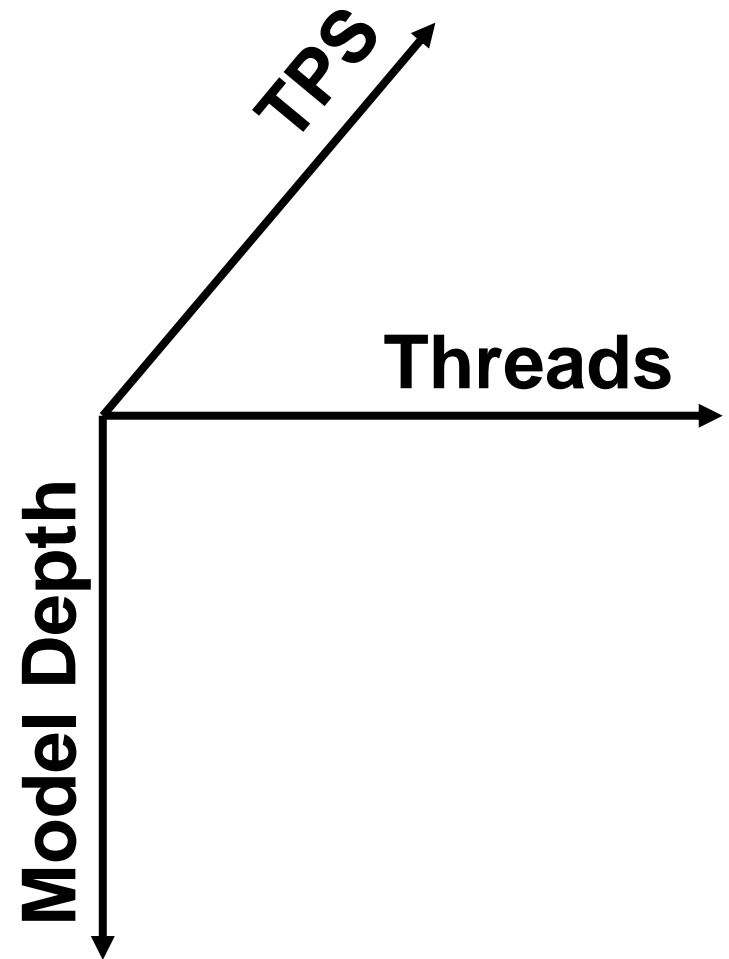


● Ref

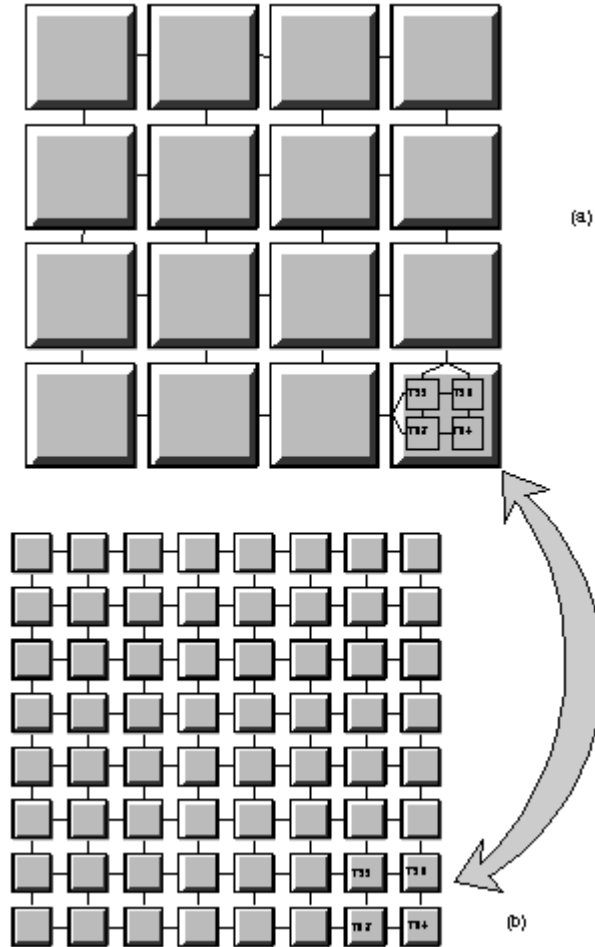
<< CAS Failure? >>



● Ref



# Shared Nothing to Scale



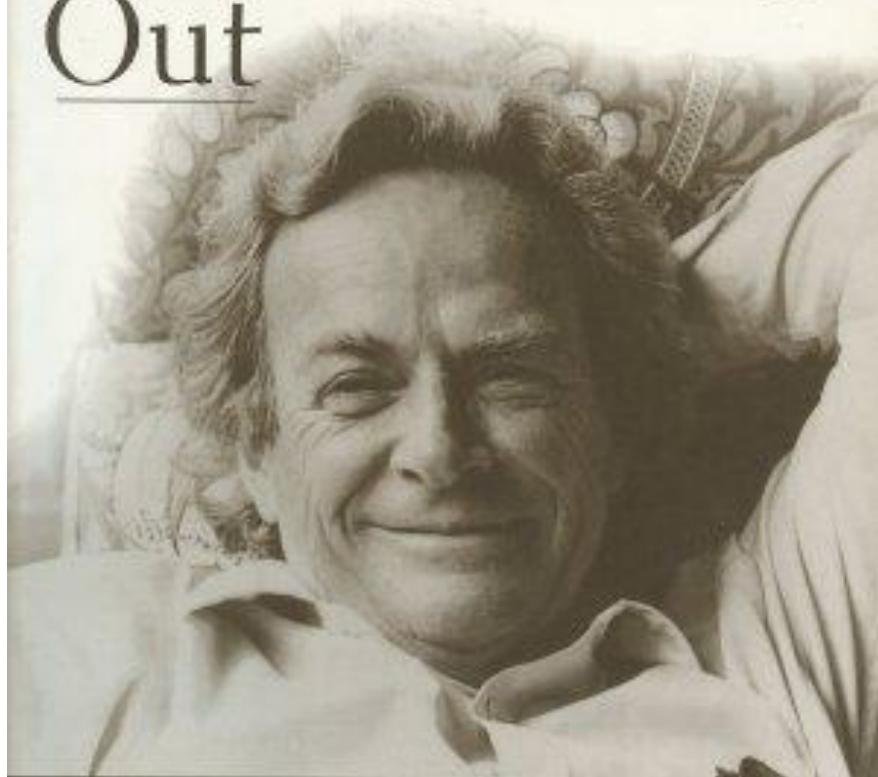
***Think Transputers  
but with message  
passing via SHM***

**4.**

***Changes in Culture***

"Feynman at his idiosyncratic, brilliant best."  
—John Horgan, author of *The Undiscovered Mind*

# The Pleasure of Finding Things Out



THE BEST  
SHORT  
WORKS OF

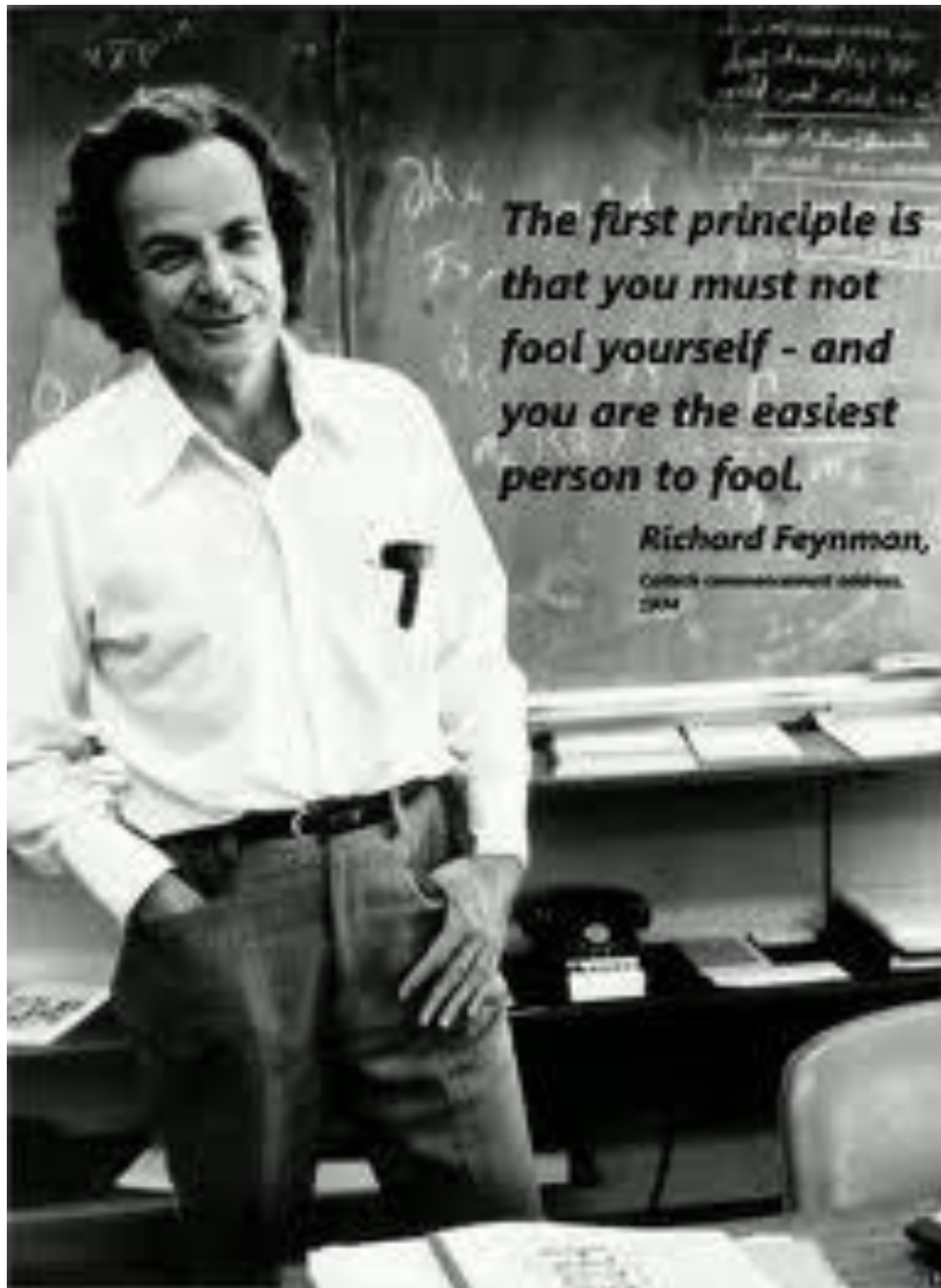
**RICHARD P. FEYNMAN**

Foreword by Freeman Dyson



# Measure Everything





*The first principle is  
that you must not  
fool yourself - and  
you are the easiest  
person to fool.*

*Richard Feynman,  
Caltech commencement address,  
1964*





# Questions?

Blog: <http://mechanical-sympathy.blogspot.com/>

Twitter: @mjpt777

***“It does not matter how intelligent you are,  
if you guess and that guess cannot be backed  
up by experimental evidence -  
then it is still a guess.”***

**- Richard Feynman**