Leveraging Legacy Assets in an Object Web World

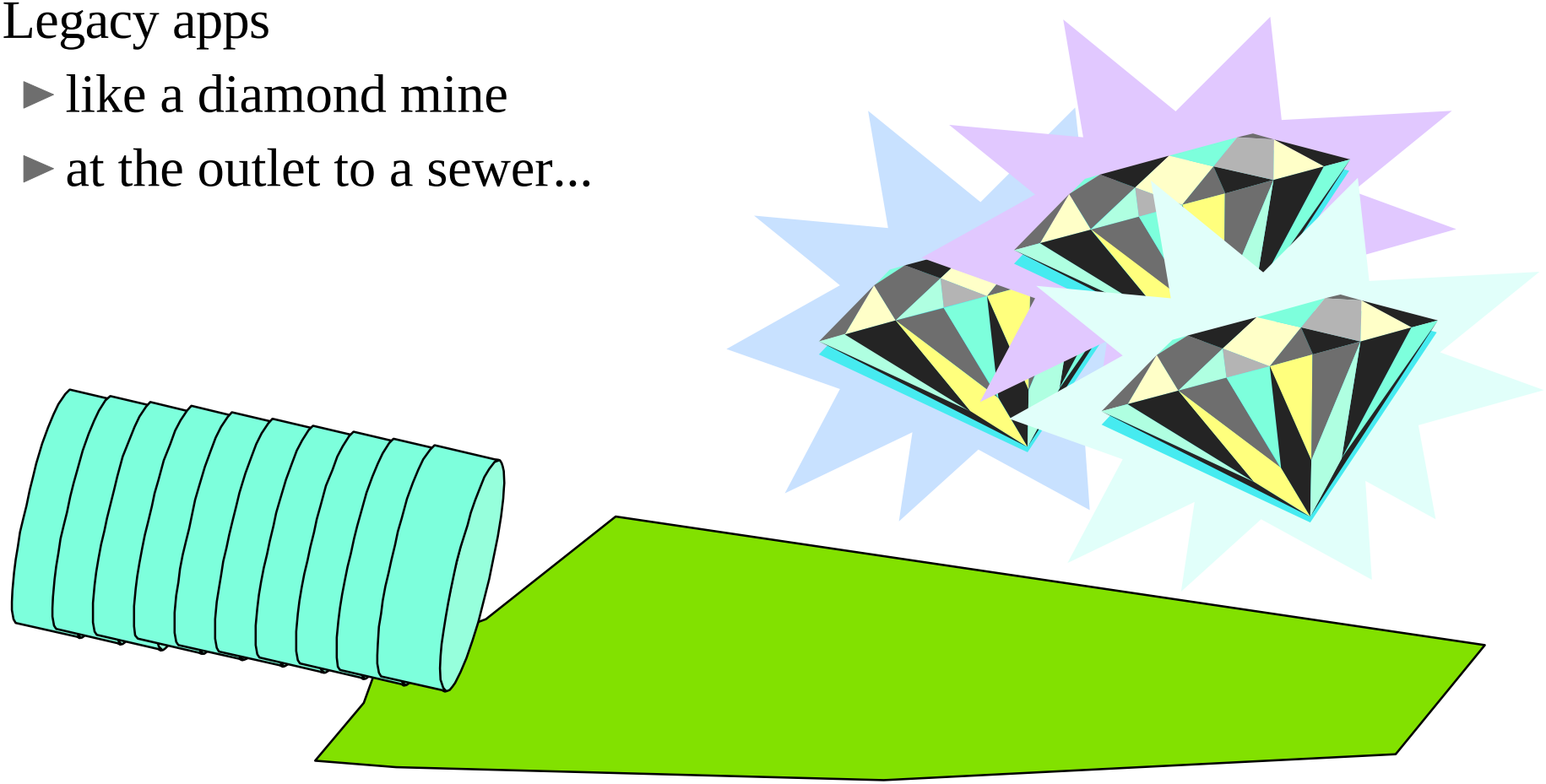
Using architecture and standards to achieve reuse (globally!)

INSPIRED

I.T. Consulting ● Training ● Research

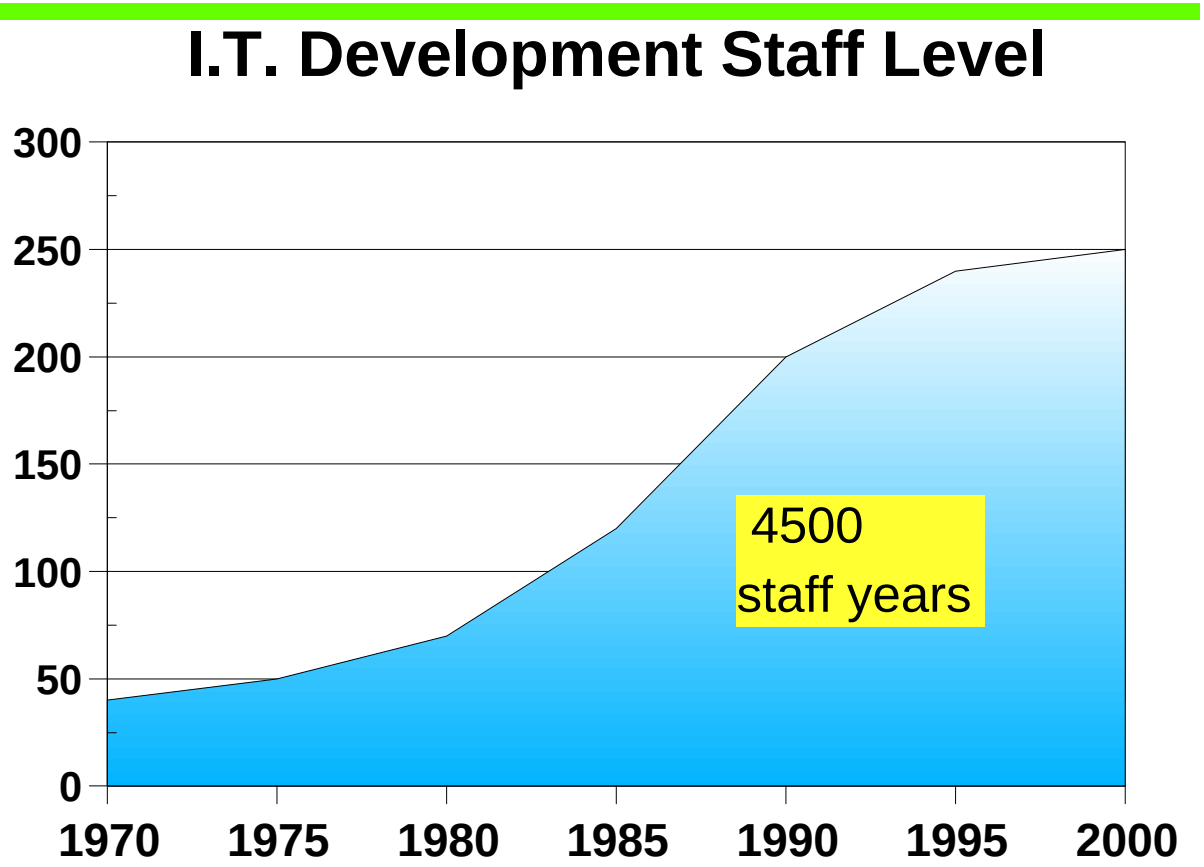
Legacy Value

- Legacy apps
  - like a diamond mine
  - at the outlet to a sewer...



The size of the problem

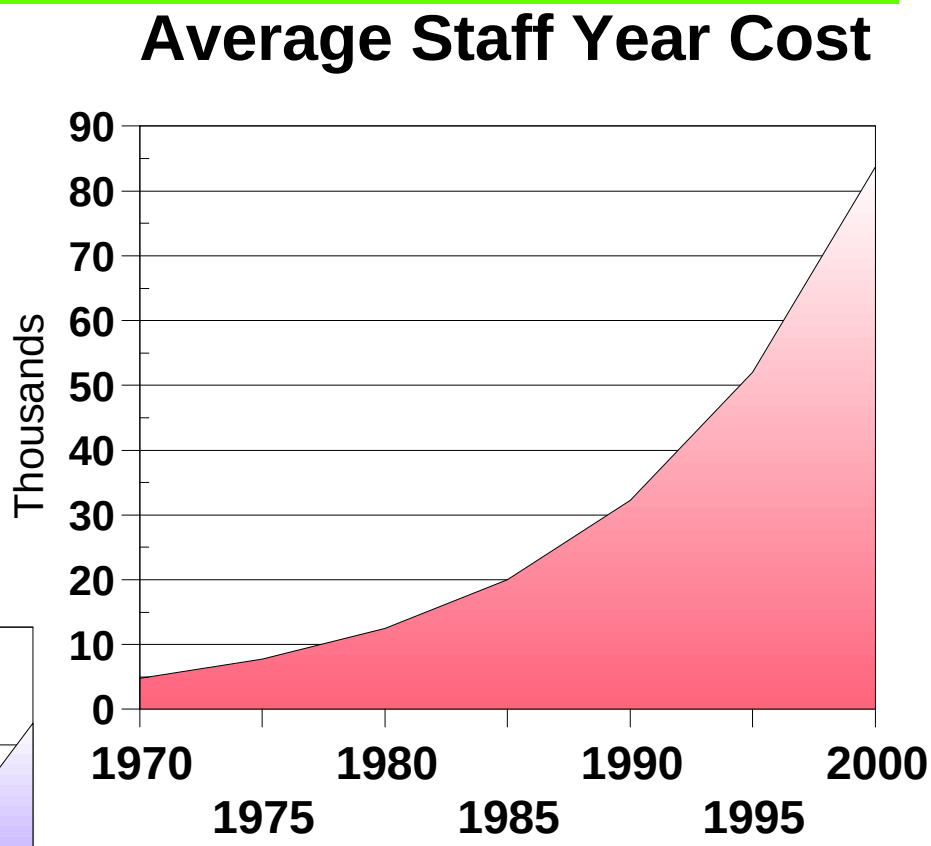
- Typical legacy systems investment of a large corporate
- At cost

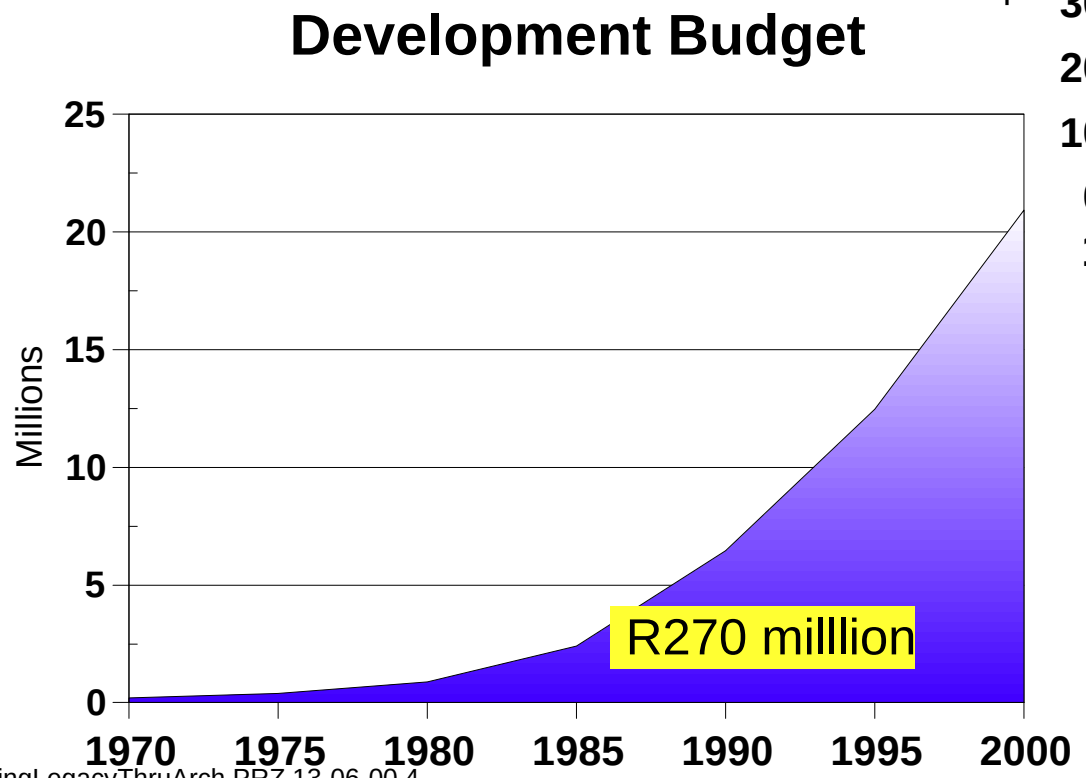


The size of the problem

Typical legacy systems investment of a large corporate

- At cost R270 million
- At replacement cost R376 million
- + Y2k
- +Line Costs





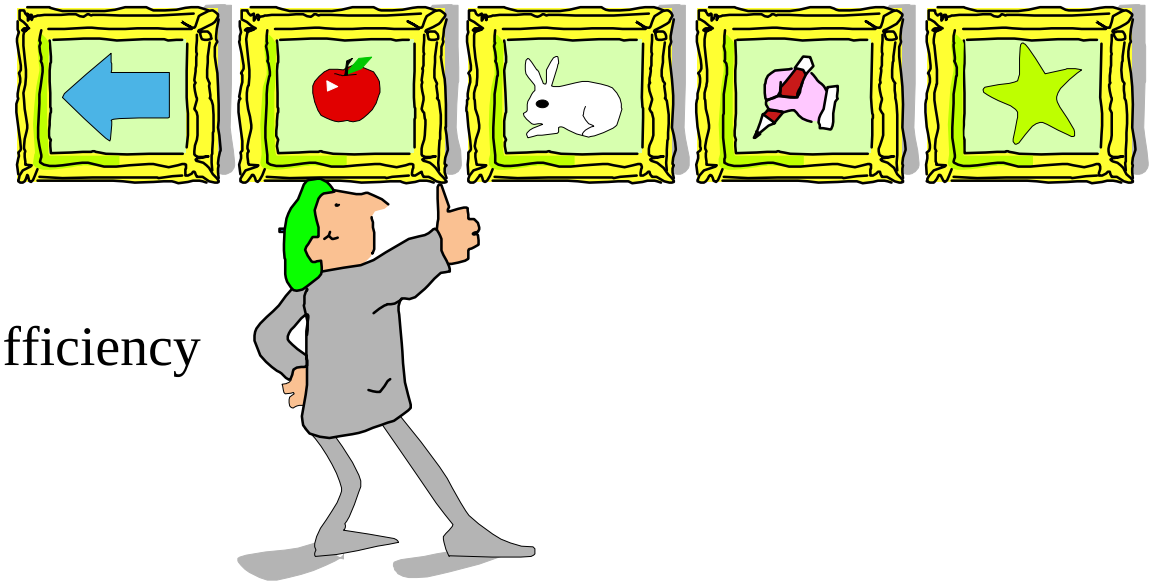
INSPIRED
I.T. Consulting ● Training ● Research

Typically cannot afford to replace everything

- Even if we want to
- And can commit the money
- We probably can't find the skills
- Or manage the projects
- Or ask business to wait long enough
- User community may also *not know* what the current systems do
  - i.e. the business process is what the systems do
  - NOT what the users manage

Prerequisites for reuse

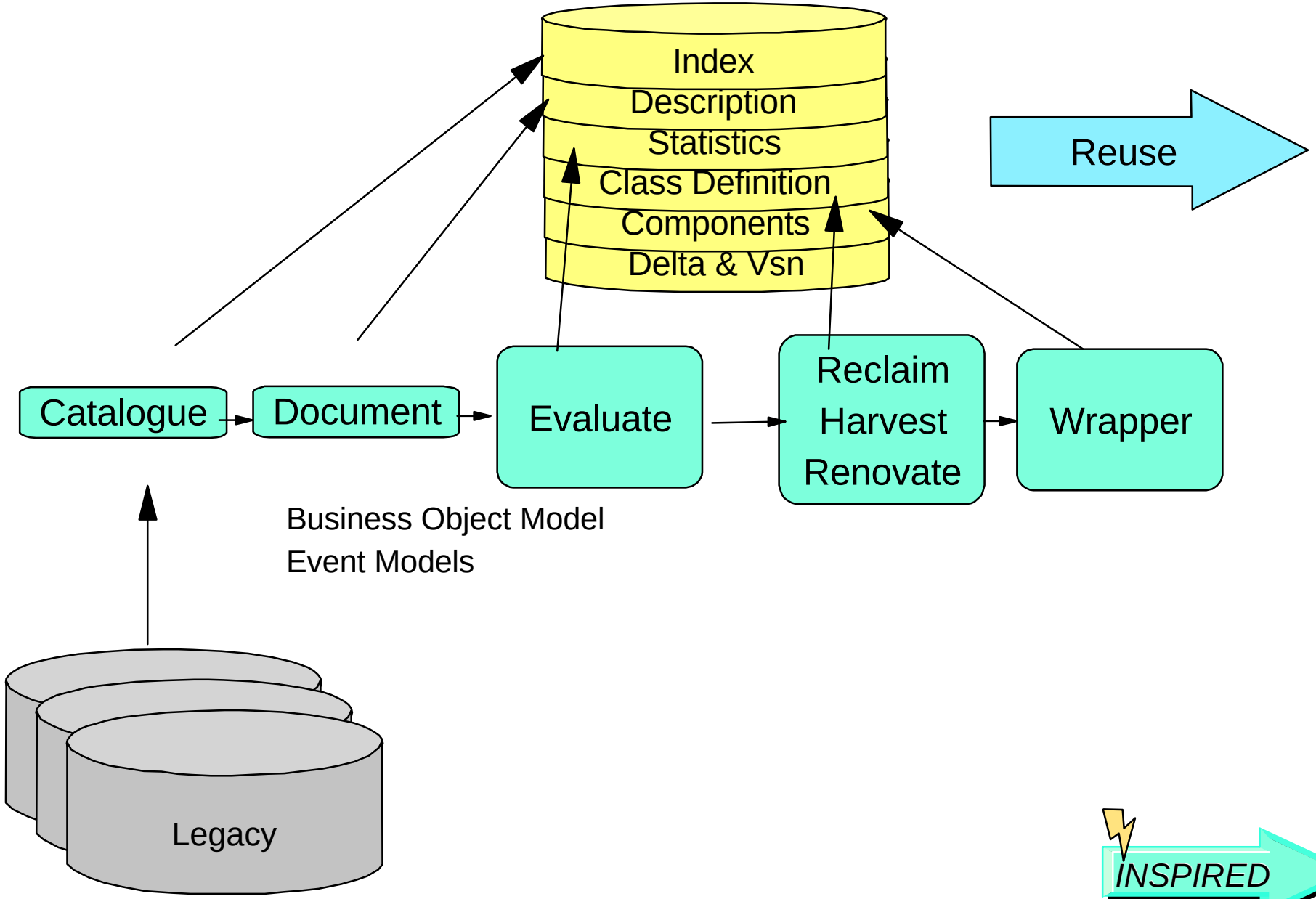
- Know what we have - cataloguing
- Know what it does - documenting
  - Structure
  - Function
  - Interface
- Is it usable?
  - Quality
  - Performance, Reliability, Efficiency
  - Portability
  - Interoperability
- Making it reusable
  - Wrappering
- Getting it used
  - Publishing
  - Incentives



What can we reuse

- Code
  - Componentize
- Data
  - Access via API, wrappered legacy applications
  - Mining, Warehouses, Marts, OLAP
  - Meta Data
- Requirements & Business Rules
  - Business Knowledge
  - Classes
  - Behaviours
- Design work
  - Frameworks
  - Patterns

Process



Where do our systems need to "play"?

- By 1997/8 Object Technology will become the predominant paradigm for conceptualizing, building and using applications.
  - *Meta Group*

- A recent survey of IT professionals concluded that "57.9% are currently developing, or plan to develop mission-critical object-oriented applications over the next 12 months."
  - *The Standish Group*

"We expect most organizations will realize the impracticality of trying to converge on a single technical architecture, and will recognize that a mixture of technical architec-tures will be the norm through 2000 and beyond. For most organizations, this realization will become apparent with the arrival of the Internet. "

— *The META Group*

Imperatives

- New technologies are more productive, but not enough for new development
- We *must* achieve reuse
  - in new work
  - *and* of existing assets
- Must achieve *interoperability*
  - with our own code
  - with industry components (widgets) and business objects
  - with parties outside our own organizations
    - Customers
    - Suppliers
    - Business Partners
    - Industry bodies
    - Electronic Markets
- Has to happen *fast*

How do we make them capable?

- In the age of the Internet, a new kind of application will predominate—an application built from new and legacy code encapsulated in software objects, running on systems both inside and outside the firewall, and interacting with each other through an Object Request Broker using defined interfaces.
  - *Visigenic*

"On average, 70% of the work for developing new applications is spent on building infrastructure, with only 30% focusing on the specific needs for the application to fulfill. A standards-based distributed computing model, such as CORBA, solves this problem by providing a reusable infrastructure that allows developers to focus on the specific business problems at hand."

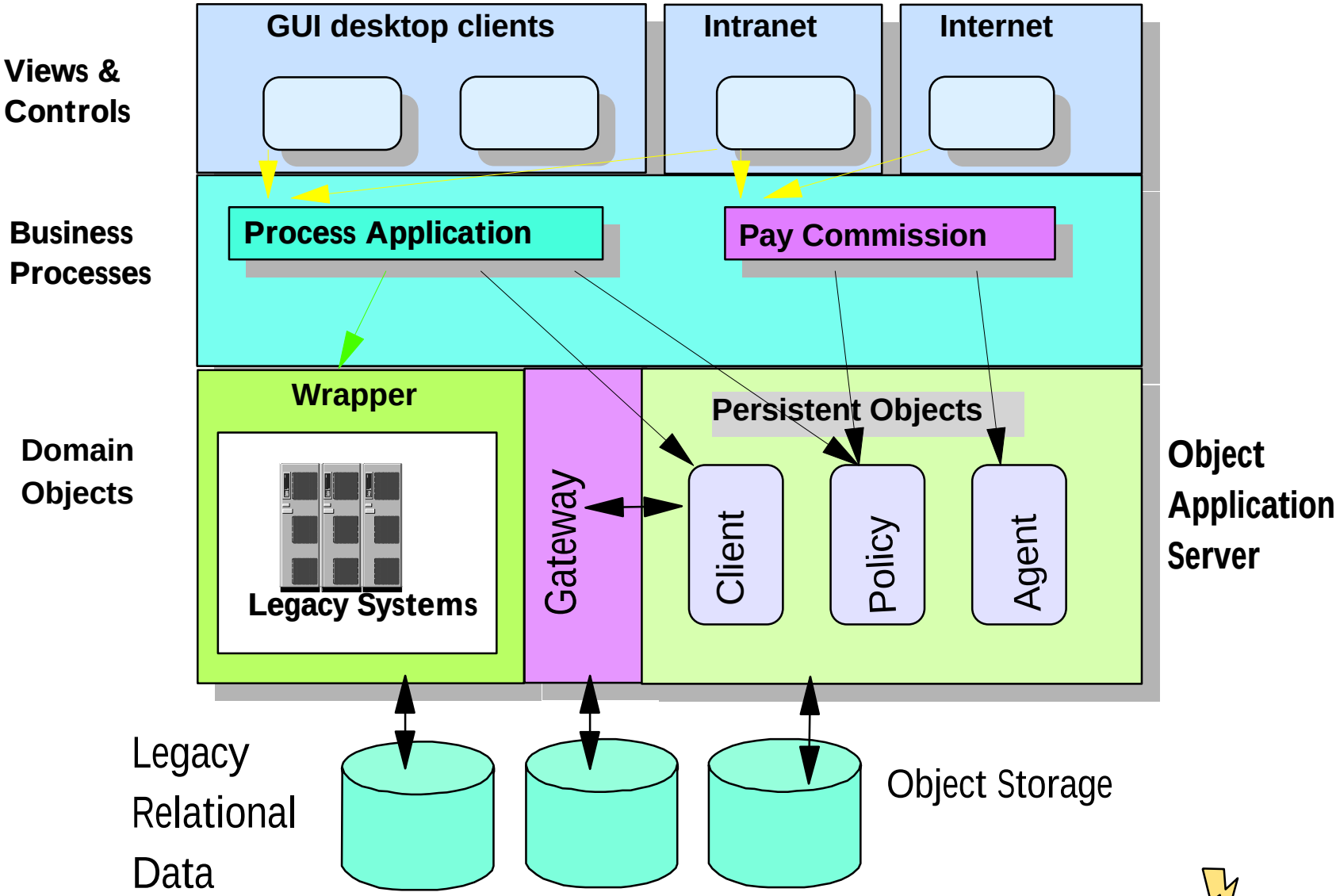
- *The Standish Group*

Before & After - Legacy Applications

- Unique API
- Unique access mechanism
- Nonexchangeable data format
- Limited metadata
- Inadequate interoperability
- Limted/proprietary security

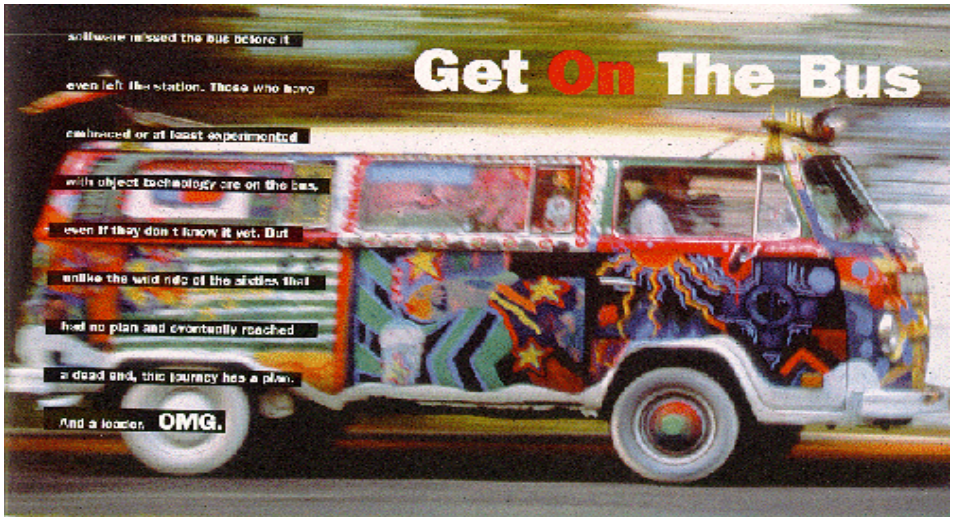
- The desired API
- Uniform Access mechanism
- Exchangeable data format
- Complete/uniform metadata
- Meets interoperability needs
- Uniform security interfaces.

Layered Application Architecture



Role of Standards

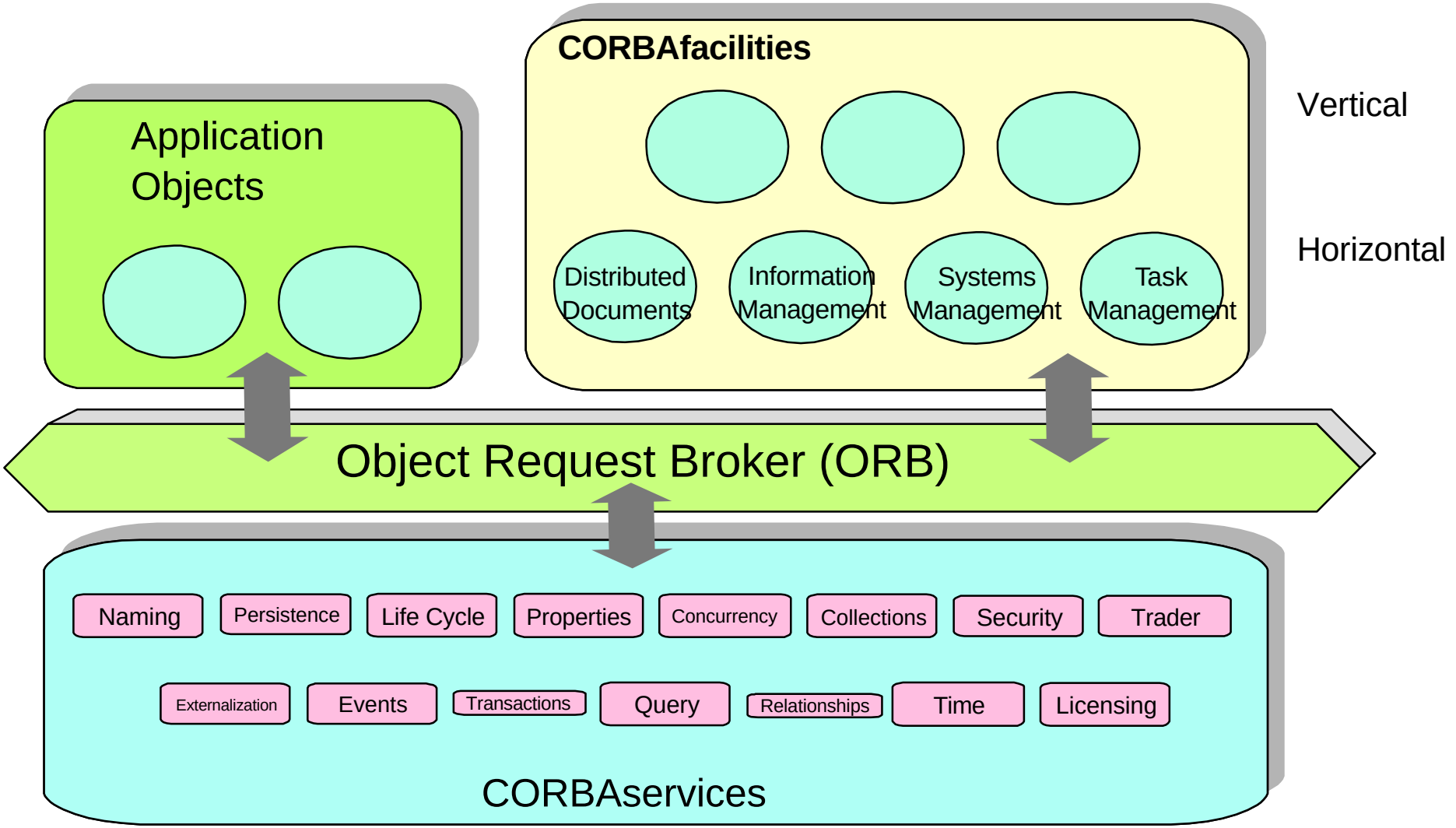
- Ideally we could develop in a language appropriate to our project and application AND
- Integrate with other applications written in other languages
- Distributed Object Standards attempt to make this possible
  - even when the objects reside in different architectures, machines and locations
- Competing Architectures
  - DCE/RPC
  - CORBA from OMG
    - SOM, DSOM (IBM)
    - DOE from Sun
    - PDO from Next
  - COM/DCOM from Microsoft
  - Java Beans/JRI from Sun



Object Management Group

- Non-profit group founded 1991
- 700+ participants including all major players
  - Microsoft a member, but not an implementor
- Define Standards for Exploiting Object Technology including:
  - Object Management Architecture
  - Common Object Request Broker Architecture (CORBA)
  - Internet Inter ORB Protocol ("*eyeop*")
  - Business Object Framework
- Not a committee based approach, but rather Tender/Proposal based

Object Management Architecture



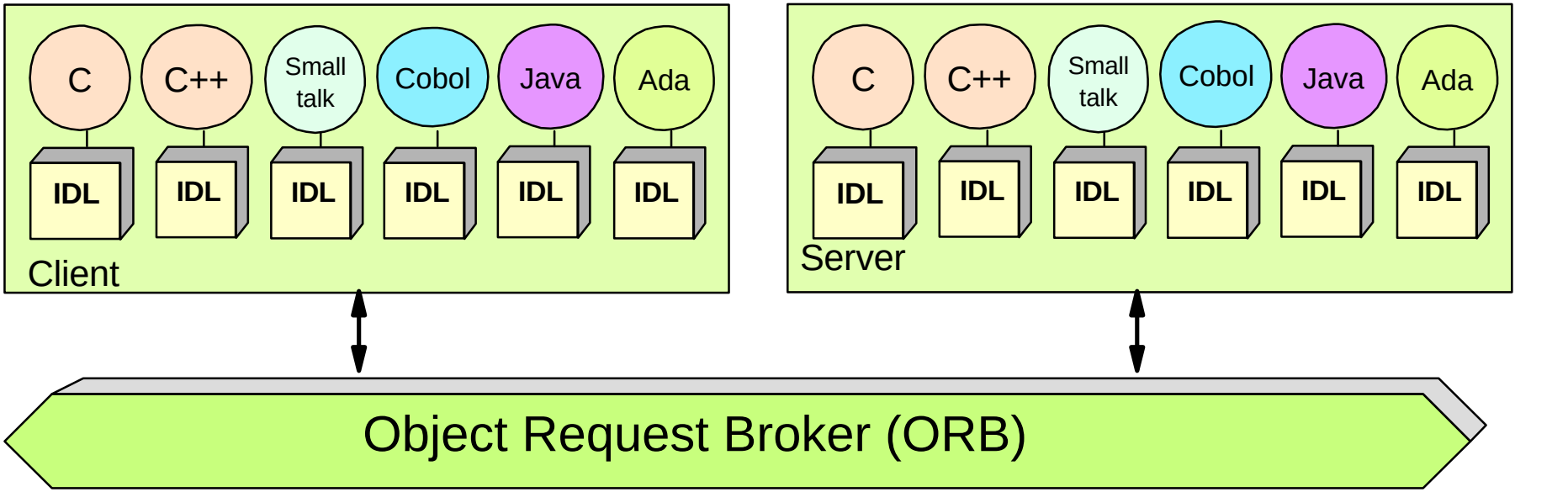
Object Request Brokers

Object Request Broker (ORB)

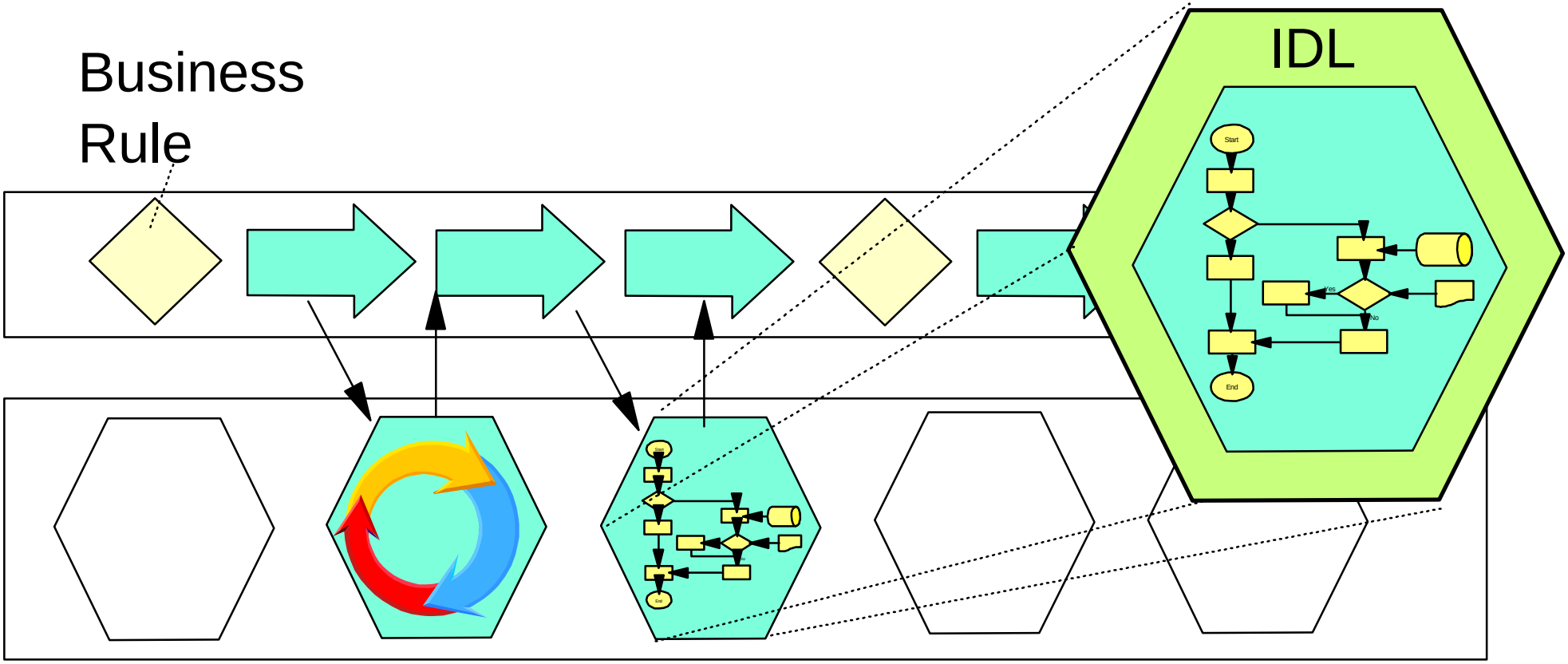
- Provides an object "bus"
- Transparent communication medium between objects
  - Locally
  - Distributed
  - Across platforms
    - Processor families
    - Languages
    - Communications Protocols
- Distributed object middleware facilities
  - Static and dynamic method invocation
  - High level language bindings
  - Self describing
  - Local/Remote transparency
  - Built in security/transactions
  - Polymorphic messaging
  - Coexistence

IDL

- **Interface Definition Language**
  - **NOT a programming language**
  - **Declarative only, implementation is vendor specific**
  - **Language and OS independent definition of APIs**
- **Grammar derived from C++ (subset) with extensions for distributed concepts**
- **Allows definition of component's attributes, inheritance, exceptions raised, typed events generated, methods supported**
- **Introspection supported**

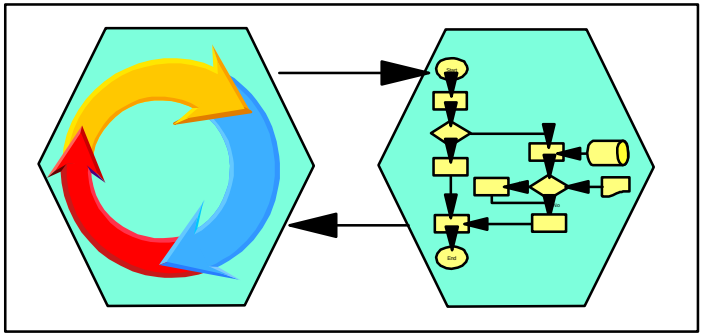


Business Process

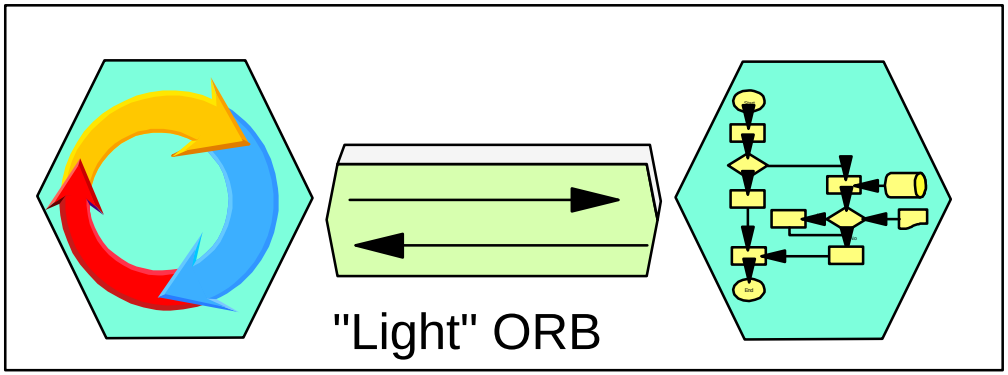


INSPIRED
I.T. Consulting ● Training ● Research

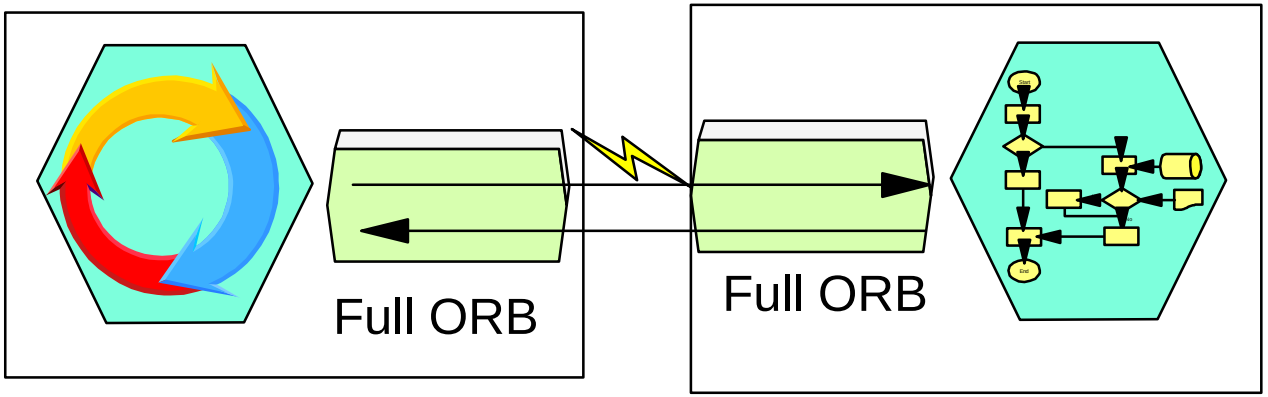
Communication between components



Same technology
Single machine

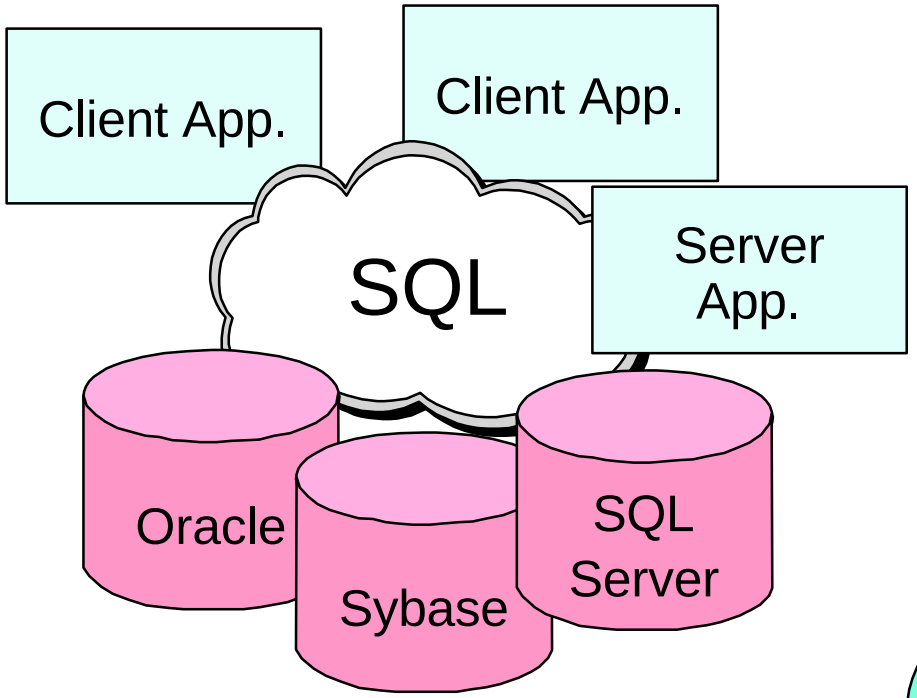


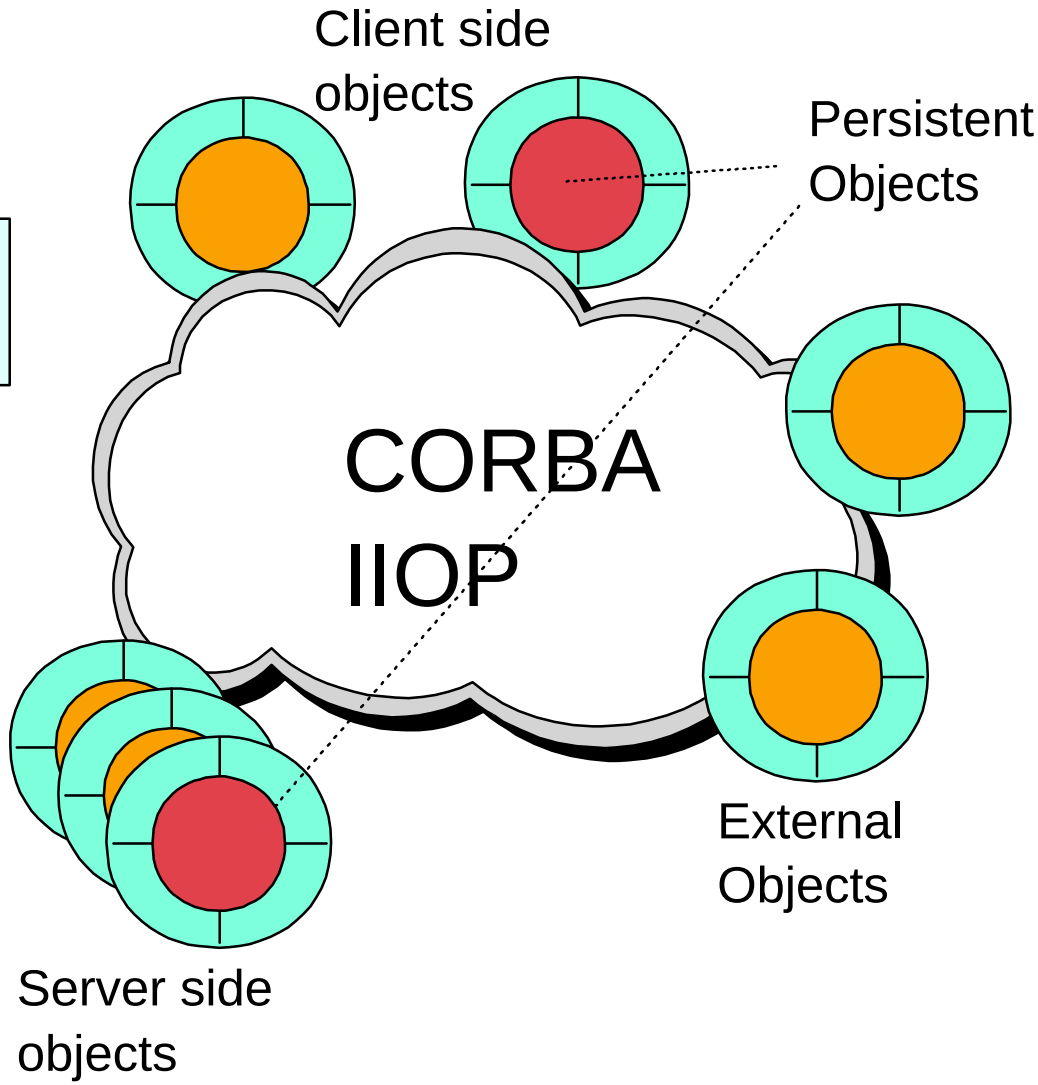
Multiple Languages
Single machine



Multiple Languages
Multiple machines

A new distributed object middleware





SQL allowed interoperability

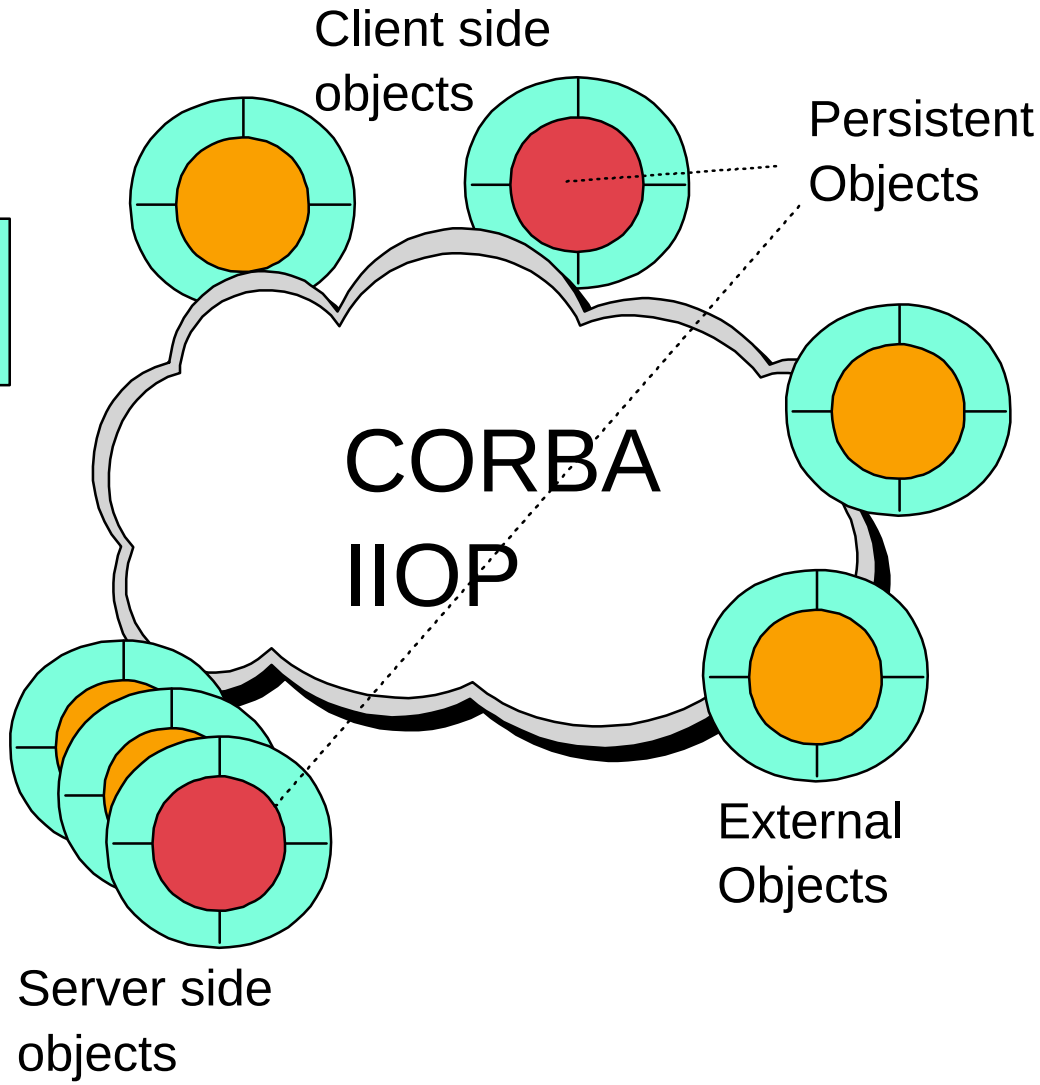
- **From various client and server apps**
- **Written in various languages**
- **To share a variety of relational databases**

A new distributed object middleware

CORBA, IDL and **IIOP** provide interoperability between objects across environments, languages and the internet.

Additionally provide a wide range of services to support distributed multi-party applications.

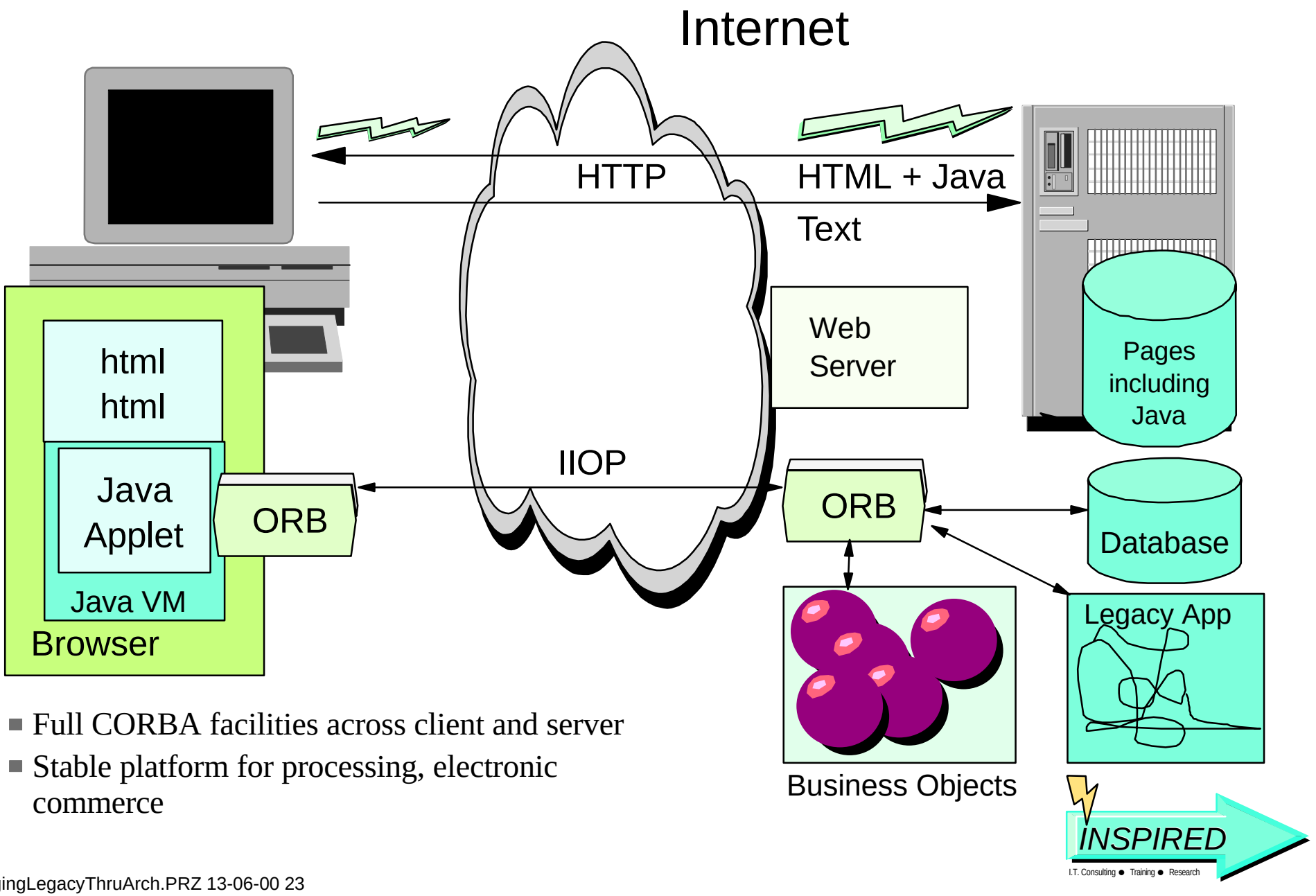
With **Java** and **Browsers**, apps can run anywhere, moving if necessary..



SQL allowed interoperabi

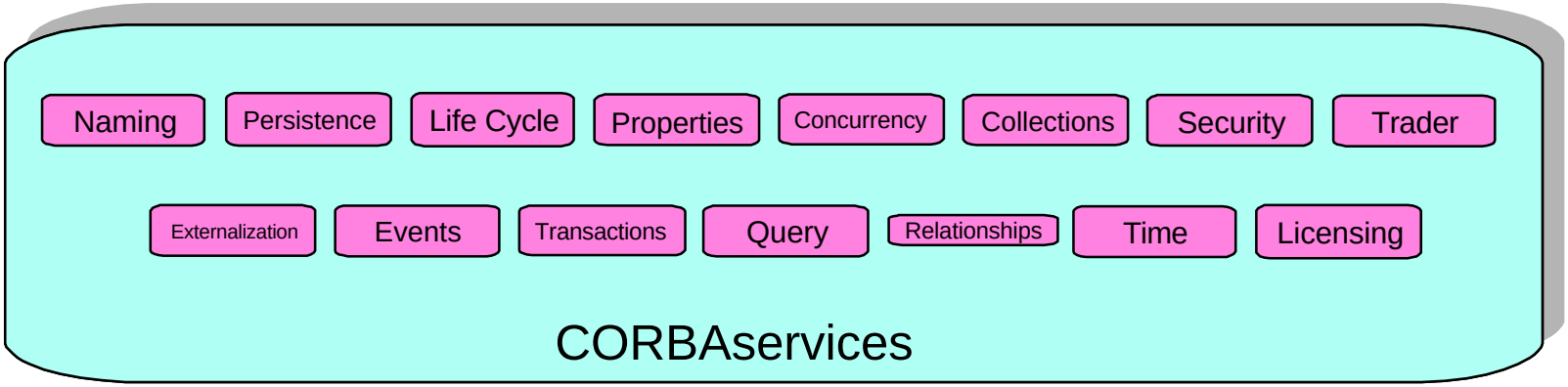
- **From various client and server apps**
- **Written in various languages**
- **To share a variety of relational databases**

Internet as Client Server Platform - 1997



- Full CORBA facilities across client and server
- Stable platform for processing, electronic commerce

CORBA Services

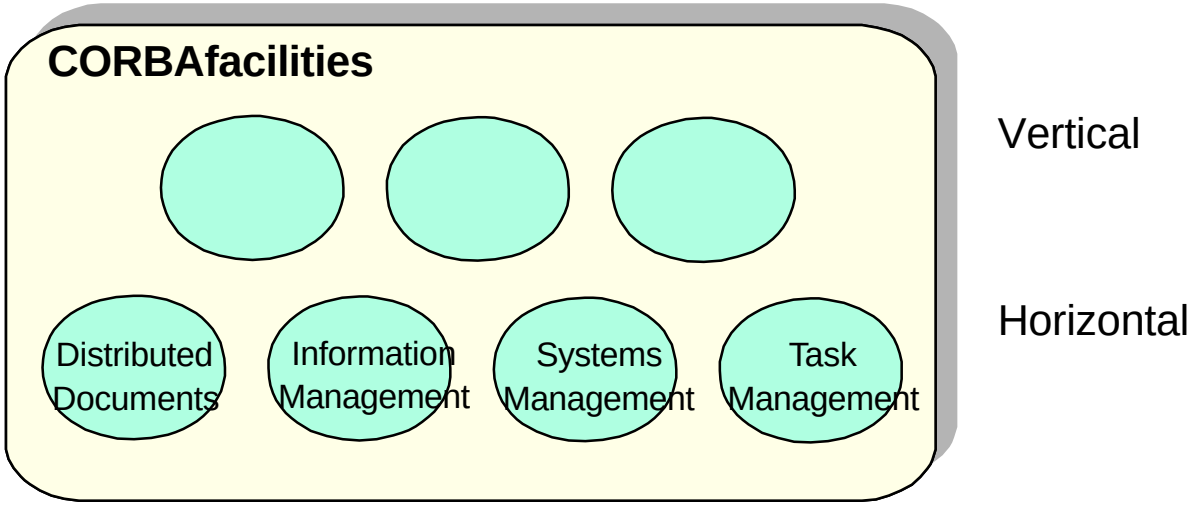


- **Lifecycle:** creating, copying, moving and deleting components
- **Persistence:** storing on object, relational database and OS files
- **Naming:** location, uniqueness across contexts, directory interface
- **Event:** register/deregister interest in events, create event channels
- **Concurrency** Control: lock mananger
- **Transaction Service**: Two phase commit coordination among recoverable components. Allows nested transactions.
- **Relationship Service**: associations between components, also across naming context
- **Externalization:** streams for import/export of component data
- **Query:** query for objects. SQL superset. (SQL3 and OQL)
- **Licensing:** metering use of components for royalty gathering
- **Properties:** associate named values with a component (e.g. current date)
- **Time:** synchronising times. Triggering events
- **Security:** authentication, access, confidentiality and non-repudiation
- **Trader:** "Yellow Pages" and bid mechanisms
- **Collection:** basic collection classes

INSPIRED
I.T. Consulting ● Training ● Research

CORBAfacilities

- IDL defines frameworks
- Provide direct support to applications
- Deal with Business Objects
- Examples: OpenDoc
  Mobile Agents
- W.I.P.



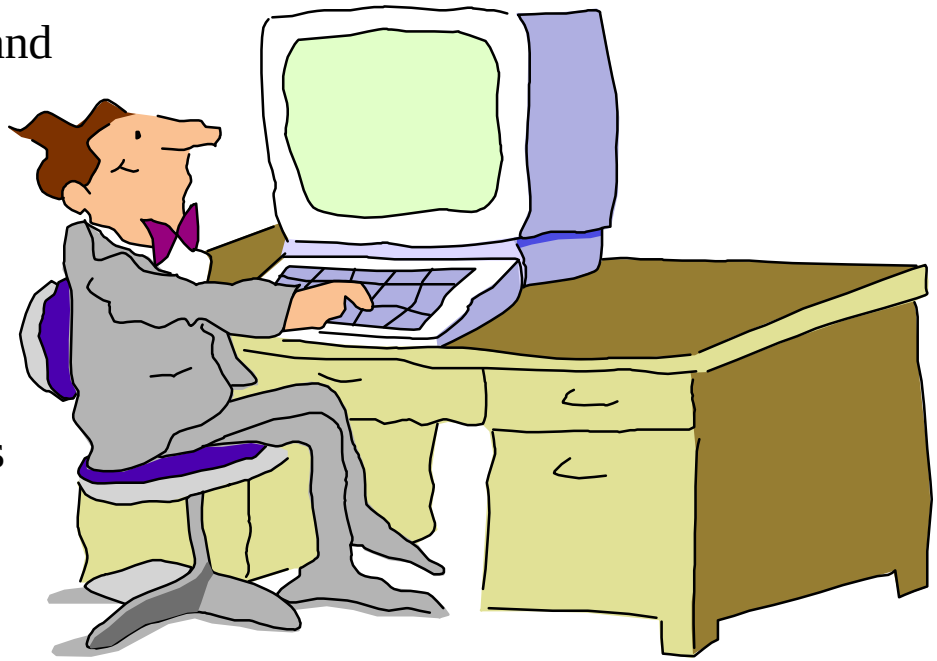
- Mobile Agents
  - Object which moves with state
  - Itinerary
  - Auto activation on arrival
  - Agent Transfer Manager

- Vertical facilities still to come
  - Customer Objects
  - Electronic Commerce
  - Health Care
  - Manufacturing etc.

INSPIRED
I.T. Consulting ● Training ● Research

Tools to help

- Restructuring tools
  - Preserve current function - improve structure and documentation
- Reclaiming tools
  - Extract Meta-data from current applications
- Reengineering tools
  - Make architecture explicit
  - Allow incremental replacement of components
- Wrappering tools
  - Provide consistent messaging interface
  - Hide implementation and quirks of legacy
- Reverse engineering
  - Reclaim design and knowledge and forward generate clean code
- Repositories
  - Collect, collate, compare, index, version, distribute, manage
- Modeling tools
  - Make assets visible, verifiable, alterable, usable!



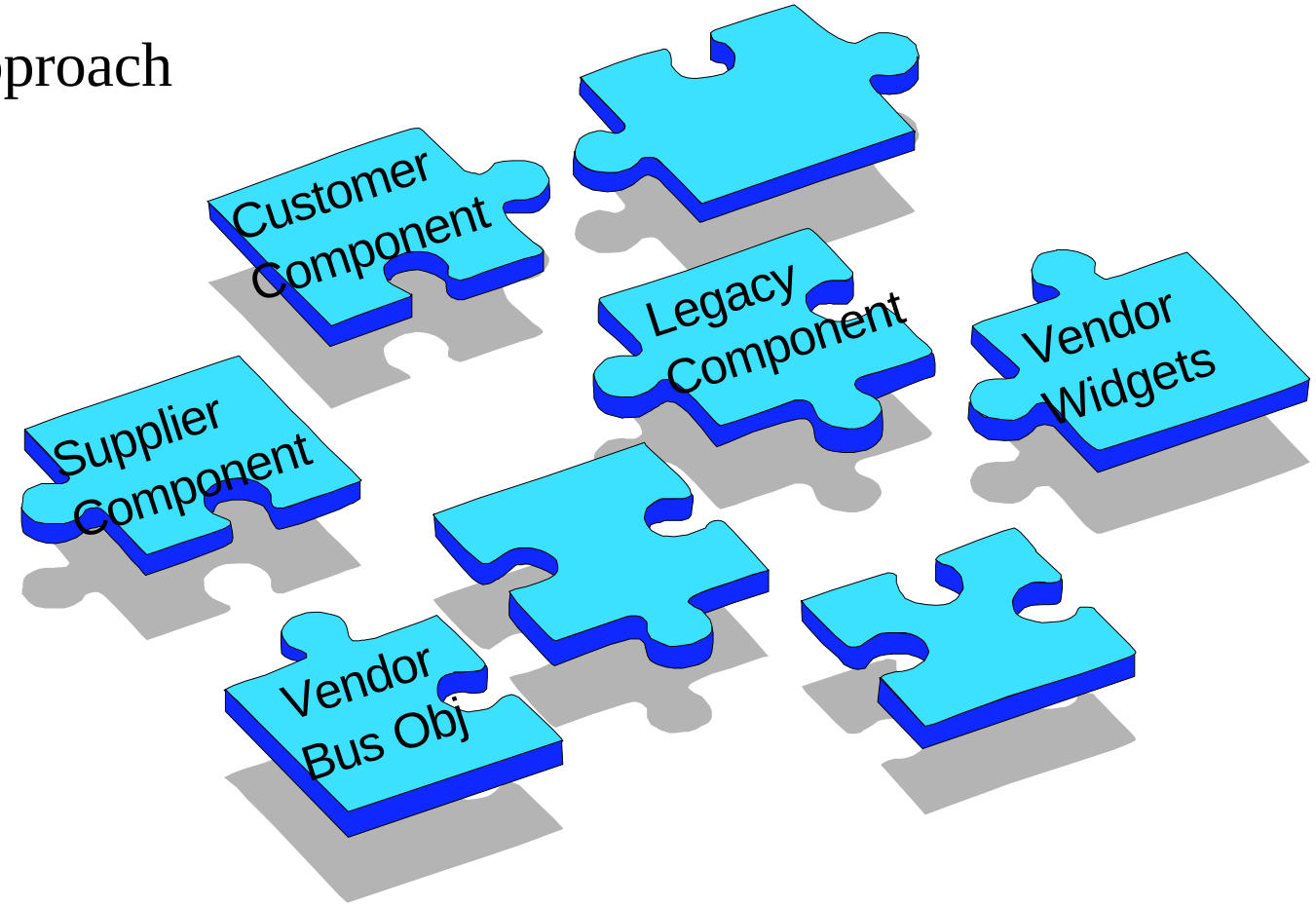
INSPIRED
I.T. Consulting ● Training ● Research

Portable Components

- Until recently, components were restricted to single environments
- That is changing rapidly
  - CORBA (OMG) provides distributed object messaging and interface standard
  - Various vendors have implementations: PDO (Next), DOE (Sun), DSOM (IBM)
  - Microsoft has countered with VBX, OCX, and Active X
  - Sun has confused everyone with Java and Java Beans
- Microsoft OLE 2 and Distributed COM are becoming real
- OpenDOC is solid and is now seeing rapid deployment

Critical Success Factors

- Architecture based approach
- Messaging is key
- Standards based
  - external
  - international
  - interoperable
  - web capable
  - secure
  - introspective
  - Open market
- IDL
- Commit high grade skills



Rewiring

- Old house with appliances directly wired into the walls
- Need to rewire with:
  - Standard sockets
  - Standard plugs
  - Earth leakage
  - May need stepup/down transformers etc
  - International standard
    - CORBA - 220 volts
    - ActiveX - 110 volts
  - Java virtual machine = travel adapter autosensing power supply
  -

Inspired

- I.T. Consulting, Training and Research
- Partners: Graham McLeod, Cornel Masson, Graham McCarthy
- Contact:
  - Postal: Box 384 Howard Place, 7450, South Africa
  - Tel/Fax: +27 21 531 5404
  - e-mail: mcleod@iafrica.com

INSPIRED
I.T. Consulting ● Training ● Research