

**ORACLE®**

# THE **INFORMATION** COMPANY

# Frameworks and Meta-Frameworks

**Duncan Mills**

J2EE Evangelist

Oracle Corp.

# Framework

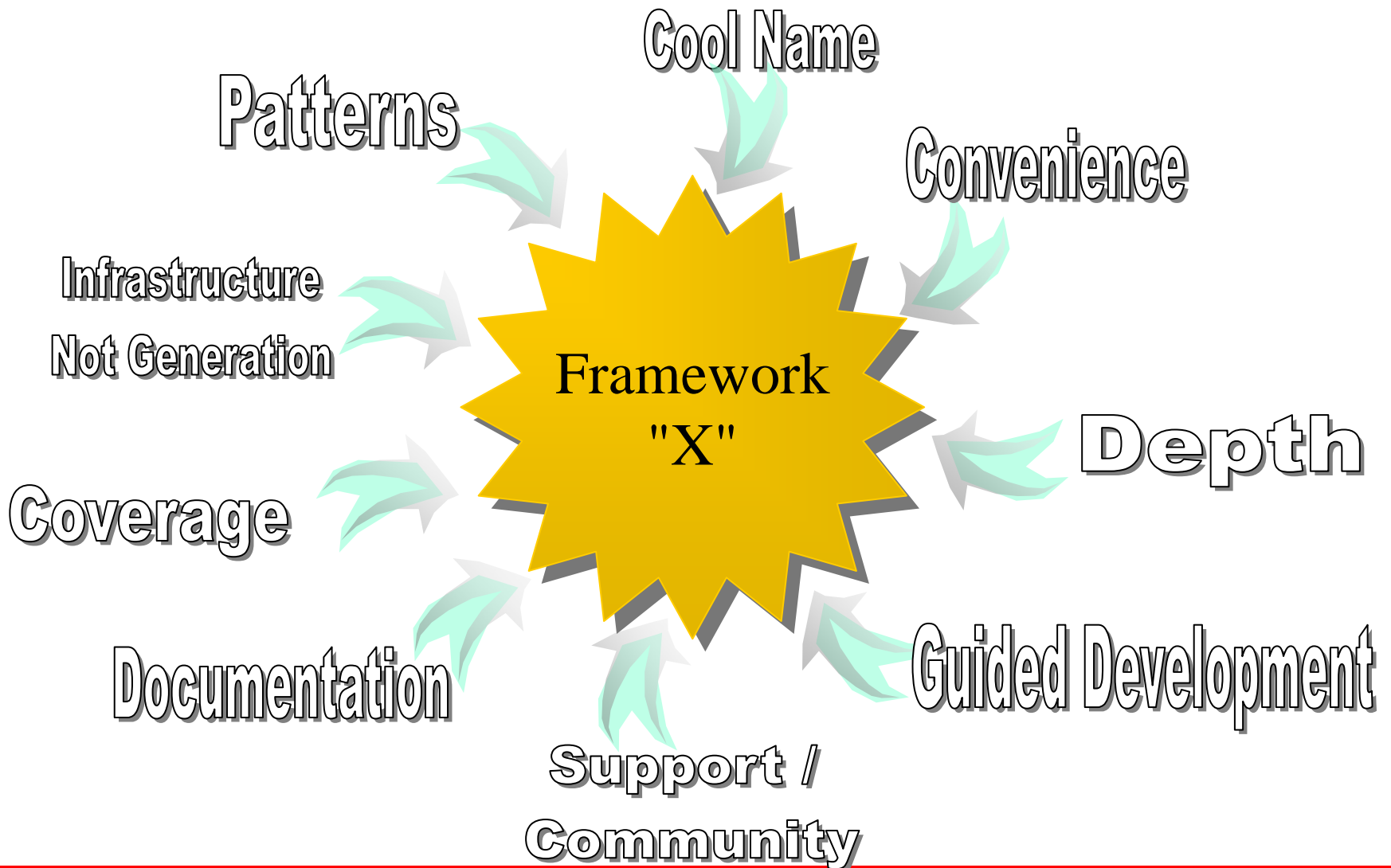
**“A defined support structure in which another software project can be organized and developed.  
...to help develop and glue together the different components....”**

- Extract from Wikipedia Frameworks definition

# What is a Framework?

- Most frameworks evolve from something simpler
  - "Scratching my own itch"
- An API is not enough
- A "one project" solution is not enough
  - That's just a toolkit

# (Good) Framework Attributes



# Frameworks

- Configured not coded
  - Focus on the things that change
- Runtime component
  - Generated code is not a framework
  - (Hey what about Annotations?)
- XML Driven?
  - No – not a requirement
- Design time component
  - From DTD to IDE

# Frameworks Are Point Solutions

- ORM
  - TopLink, iBatis, ADF Business Components, Hibernate
- Web view generation / controller
  - Too many to mention
- Binding
- Security
- Templating
- ...

# The Requirements Gap

- Frameworks evolve on top of the standards
  - Hide the dumb stuff
  - Add convenience
  - Plug the gaps
  - Solve *real* problems
- Examples
  - Struts on top of the Servlet & JSP APIs
  - TopLink on top of JDBC
  - JSF-Security on top of JSF

# Then The Standards Catch up...

- JSF supersedes Struts
  - Shale builds on top of JSF and the cycle continues
- EJB 3.0 supersedes TopLink and Hibernate
  - The 80 / 20 rule
  - But both continue to add value there is more for the spec to do
- Frameworks *should* get thinner
  - Fade away into implementations

# Meta-Frameworks

- A framework for frameworks
- Convergence on a platform
  - Addressing the whole problem
- Key Differentiators
  - **Scope** - end to end
  - **Plugability** – incorporate a selection of point solutions
  - **Philosophy** – Consistency in approach
  - **Ownership** – pulling the pieces together

# Meta-Frameworks: Scope

- All layers involved or encapsulated
  - Conventional MVC
  - Asynchronous integration
  - Cross cutting concerns
    - Security & identity
    - Logging and management
  - Plugging the gaps (again)
- Out of container
  - Desktop applications

# Meta-Frameworks: Pluggability

- Meta-frameworks consume and enhance
  - May provide some infrastructure
  - Will provide a lot of adaptors
- Should provide choice of implementation
  - e.g. ORM solution
- Abstract out the implications of that choice
  - Implicitly loose coupled
  - Firewalling
- Future proof

# Meta-Frameworks: Philosophy

- Working in an consistent manner
- Adapt to changing circumstance
  - Customer requirements
- Generally framed around a key integration component
  - e.g. Spring: the bean factory
  - ADF: ADF Model Binding

# Meta-Frameworks: Ownership

- Commercially backed
- Longevity
  - APIs change → the framework adapts
  - Oracle Forms as a case in point
- Stability (without stagnation)
  - Components in lock-step
  - Certified combinations
- Tooling
  - Key to widespread adoption

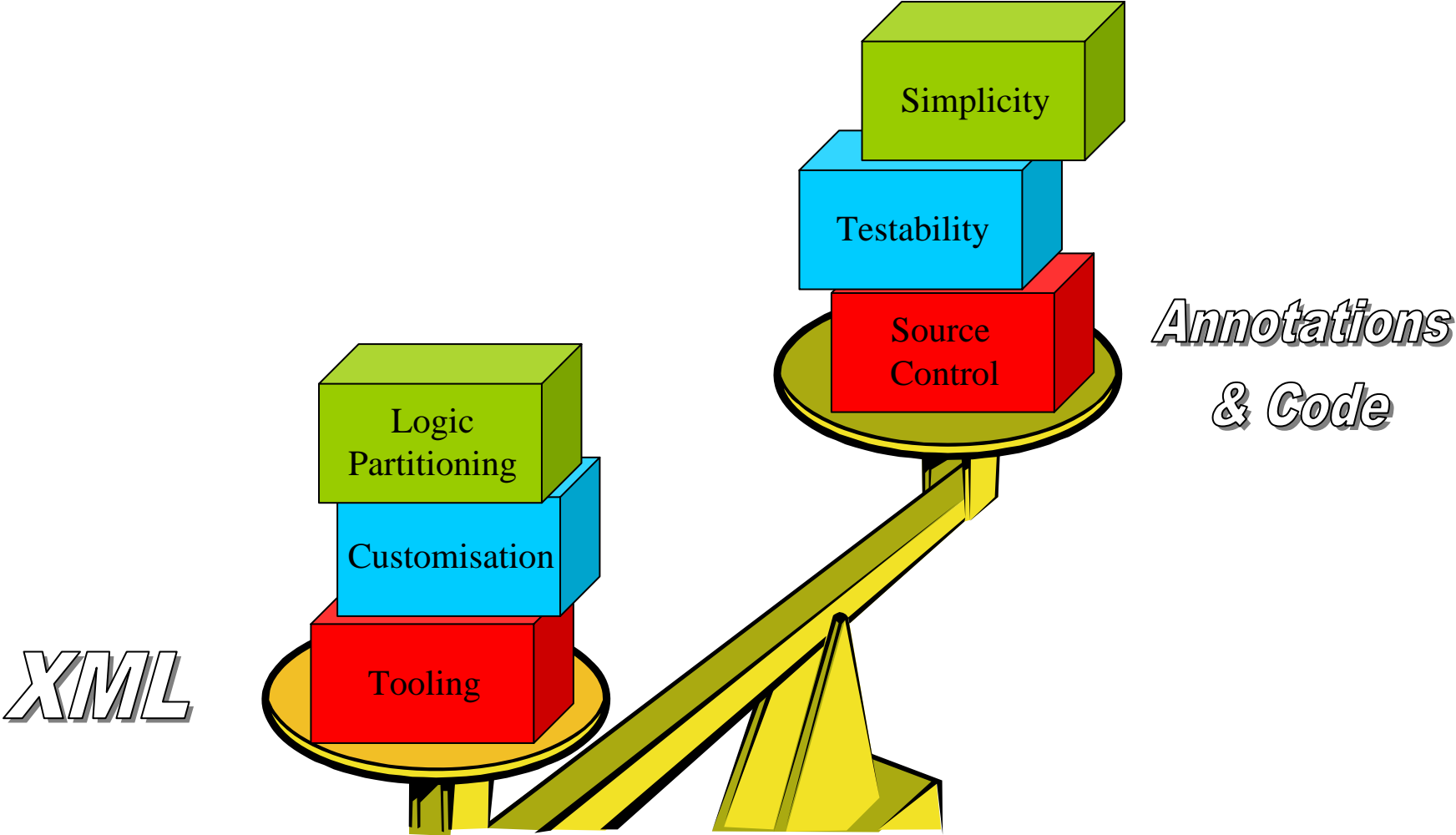
# Are There Any Meta-Frameworks?

Candidate	
Oracle ADF	A-
<b>Spring</b>	A-
<b>Keel</b>	B-
JBoss Seam	C-
Apache Beehive	C-

# Framework Challenges

- The configuration problem
- Debugging
- The evolving nature of applications

# The Configuration Problem

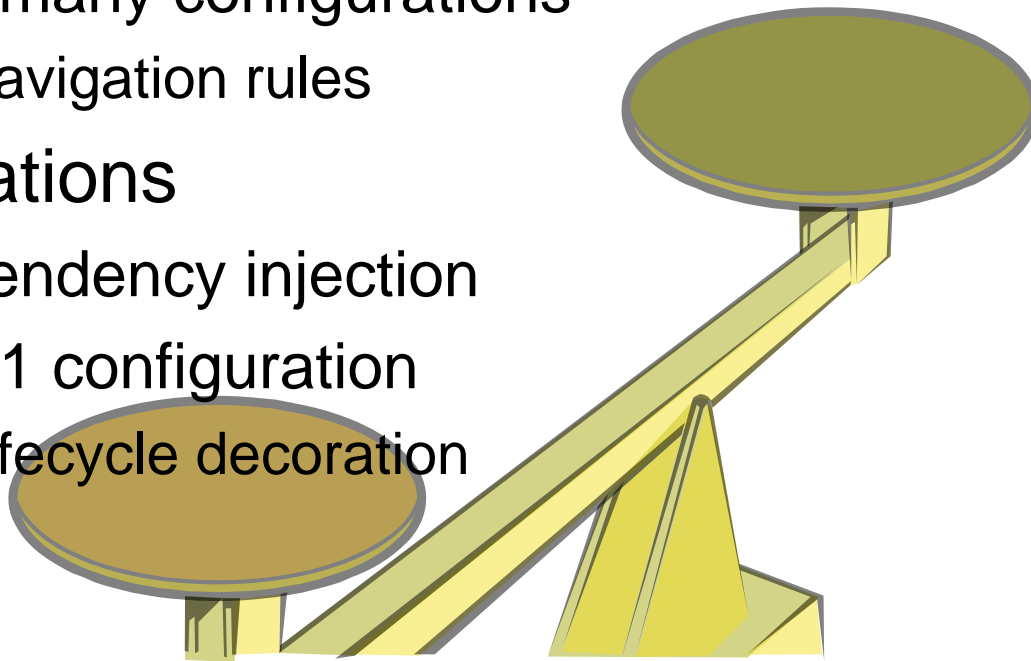


*XML*

*Annotations  
& Code*

# Striking the Balance

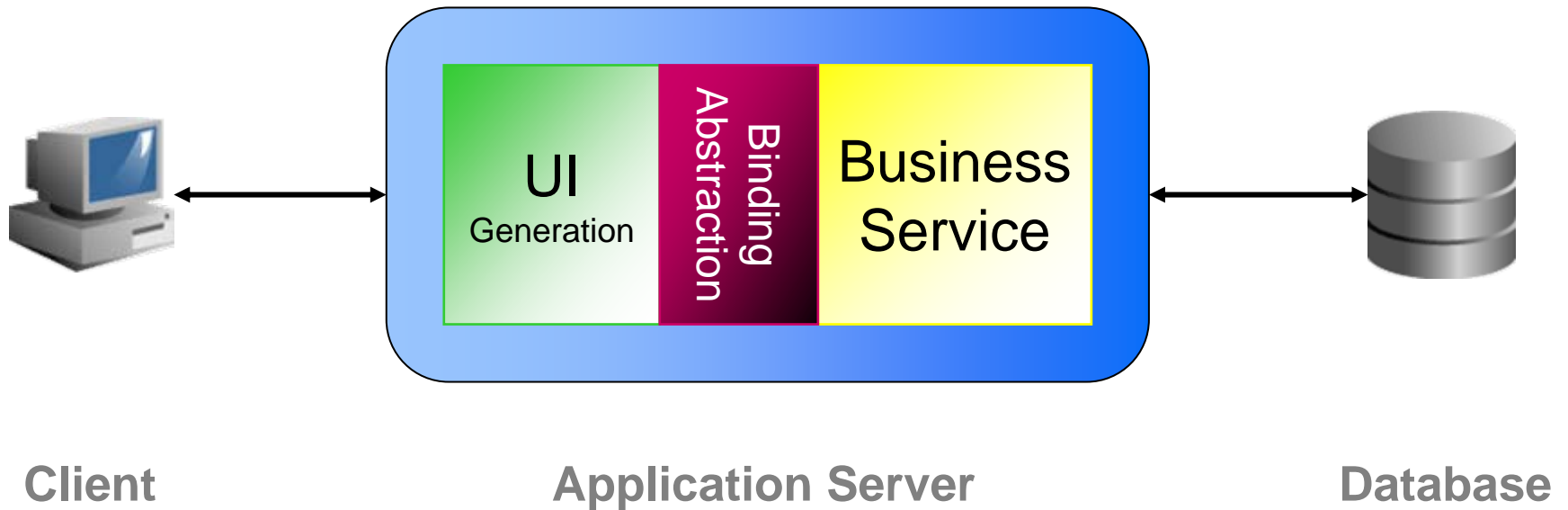
- XML
  - Customisable elements
  - 1-to-many configurations
    - Navigation rules
- Annotations
  - Dependency injection
  - 1-to-1 configuration
    - Lifecycle decoration



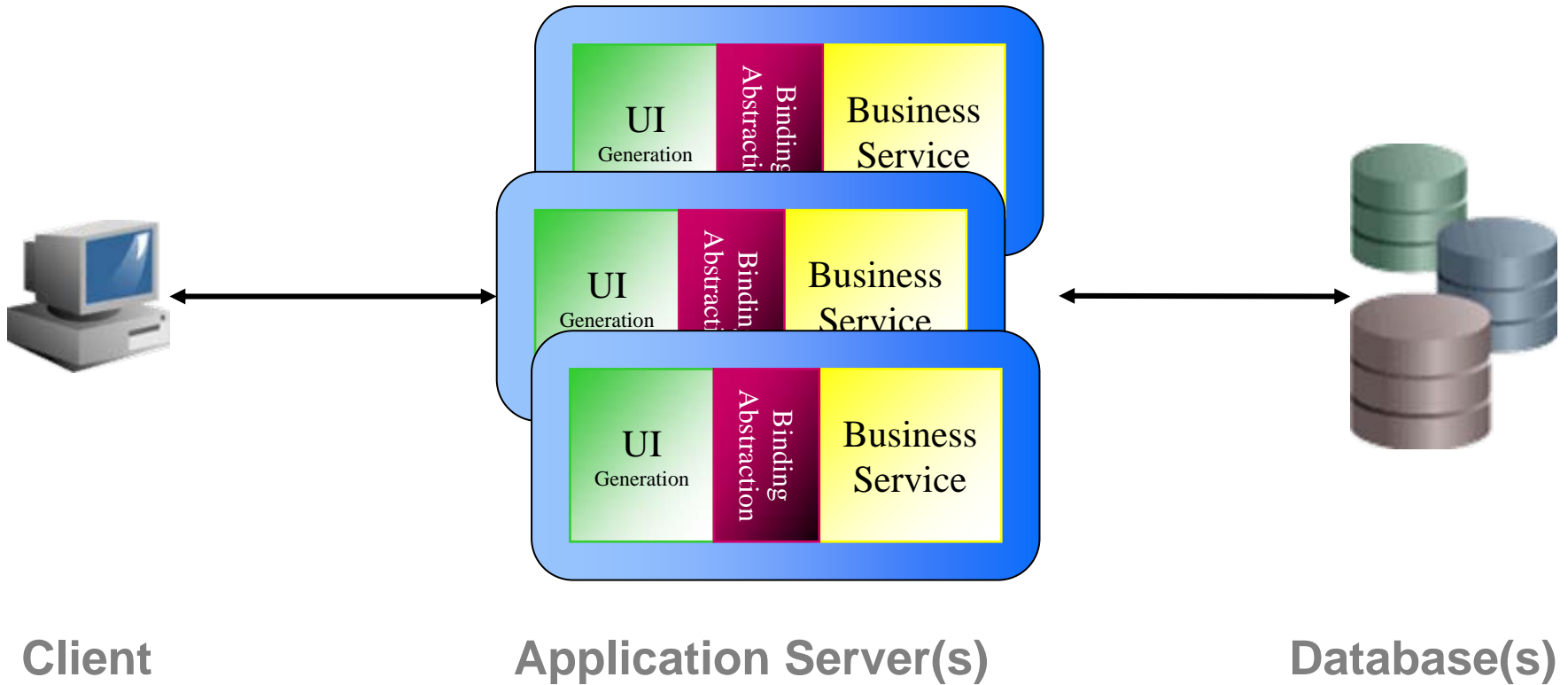
# The Debugging Problem

**“Framework users have to know all about debuggers. Debuggers, however, have yet to return the favour”**

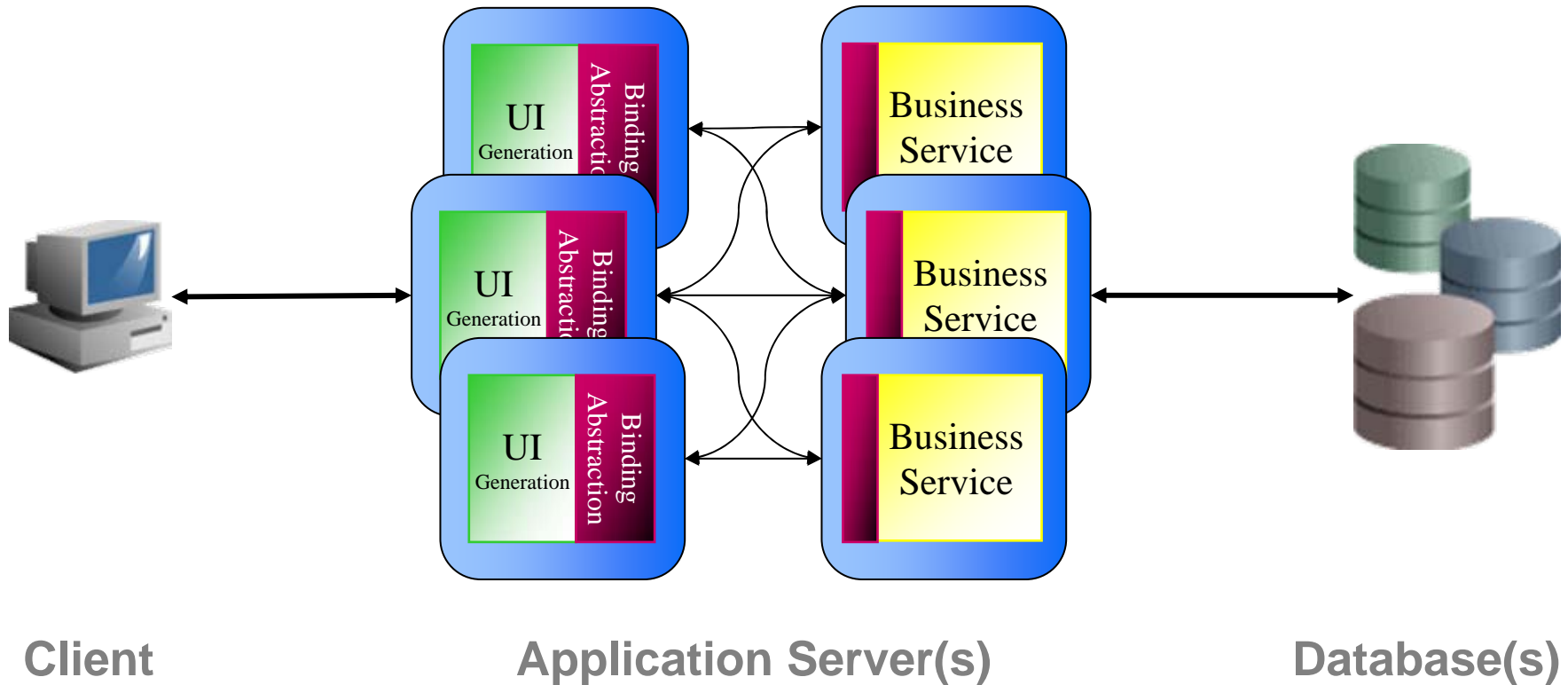
# A Typical Application?



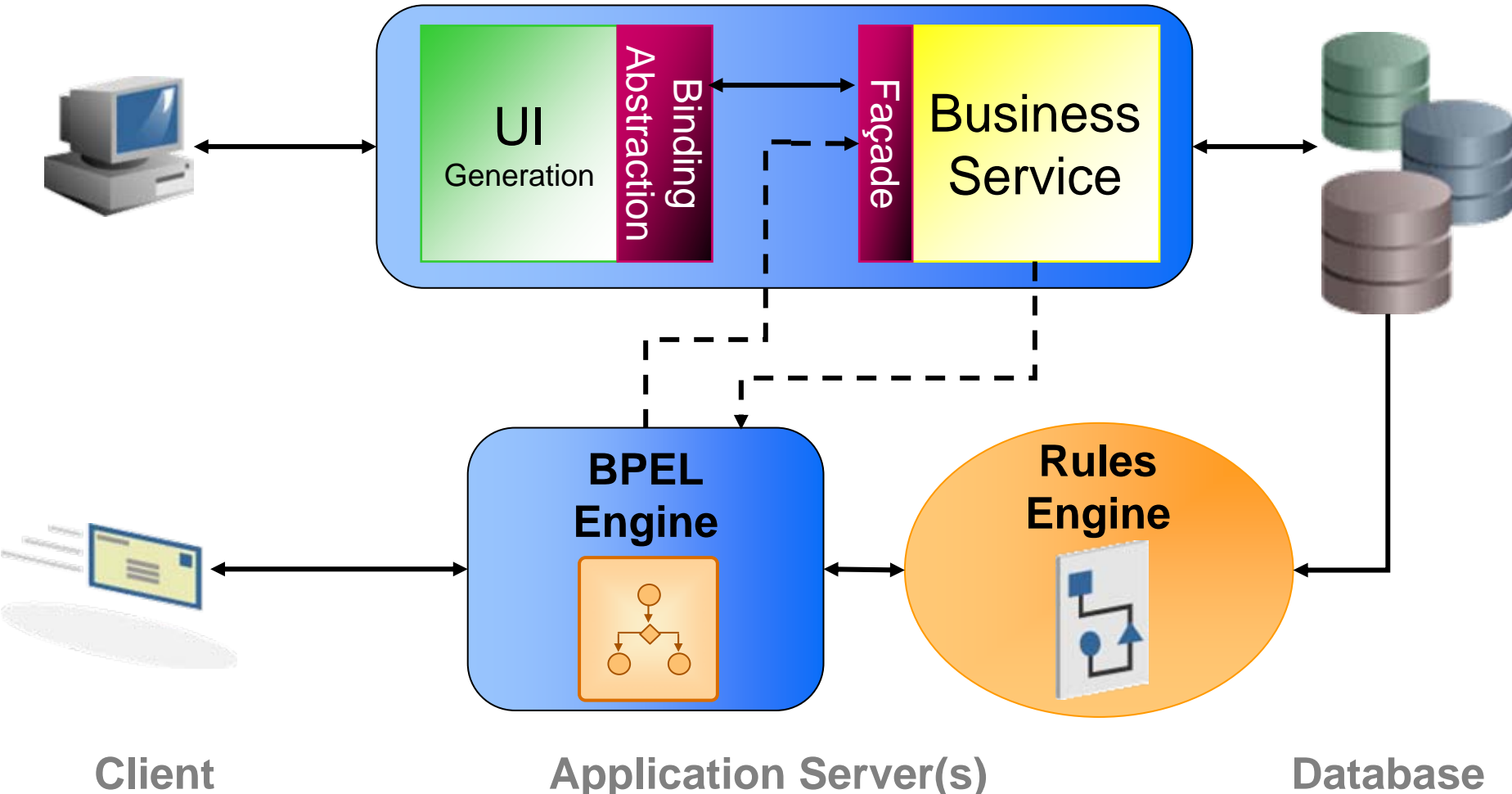
# More Realistic?



# More Realistic?



# How Things Really Are



# Wrap-up

- Standards are maturing
  - The role of frameworks is changing
- Meta-frameworks become essential services
  - Integration
  - Tooling
  - Monitoring and management
- The next generation of challenges will be much harder
  - Abstracted Debugging
  - State management

**ORACLE®**