Architecture of a Business Framework for the .NET Platform and Open Source Environments

Thomas Seidmann

Cdot AG, Wilen, Switzerland

Outline - Technical Part

- Genesis and audience.
- Application architecture.
- Framework requirements.
- Components of the framework.
- Extensibility.
- Testing provisions.
- Considerations for Open Source Systems.

Outline - Philosophical Part

- Closed Open Source model and its challenges.
- The “Problem” with customers and its solution (fragments of a business model).
- Virtual Company and Real People.

Genesis and Audience

- Side effect of a proof-of-concept.
- Based on the .NET platform.
- Now part of a broader product Cdot-InSource (CIS).
- Not a RAD tool.
- Aimed at software engineers as developers.

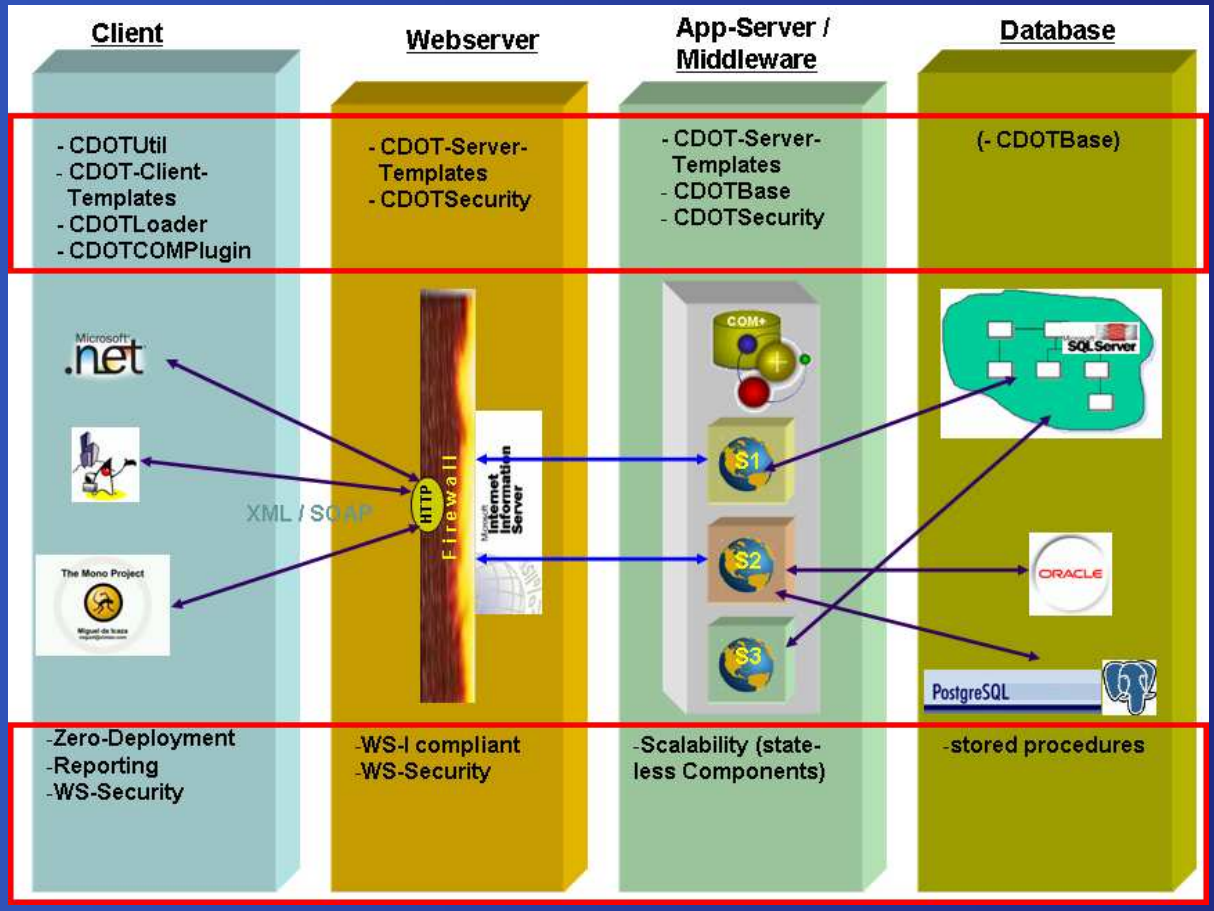
Genesis and Audience (cont'd)

- Provide developers with samples, blueprints and templates.
- Efficient track to building "Enterprise"-style applications.

Applications Architecture

- Multi-layered applications.
- Web service centric approach.
- Typically centered around DBMS.
- Mainstream for UI: zero-deployed “smart clients”.

CIS Applications Architecture



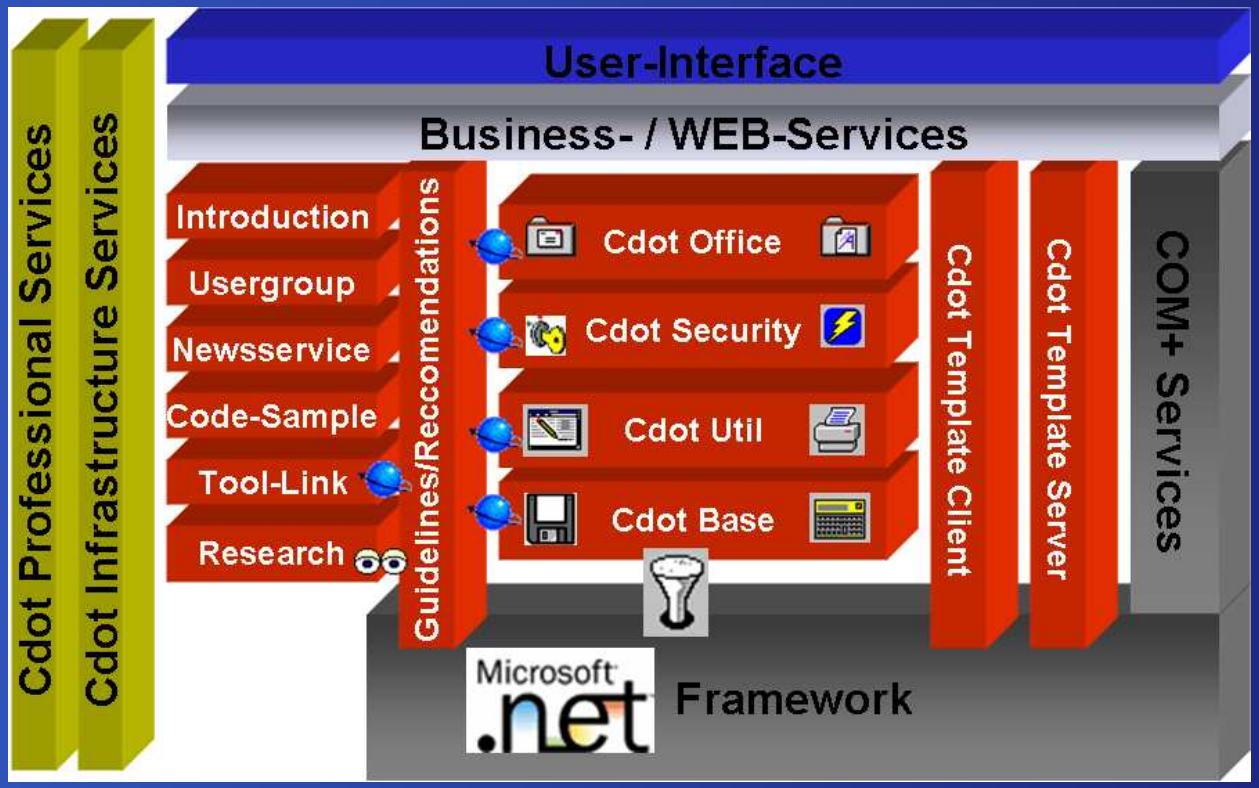
Applications Architecture - Outlook

- Service orientation (message passing) vs. current RPC style.
- Peer-to-peer networks.
- Grid computing.
  - Distributed shared memory.

Framework requirements

- Adhesion to and support for the desired architecture.
- Provide Single Point of Reference (SPoR).
- Provide missing pieces of the picture.
- Extensibility.
- (Compliance with MONO for OSS.)

CIS Framework Components



Communication related components ...

- ... Middleware
- Security issues of client to web service communication:
  - Authetication: WS-I BP compliant, that is WS-Security.
  - Privacy and integrity: WS-SecureConversation.
  - Authorization: own RBAC-based authorization subframework.

Communication related ... (cont'd)

- Communication between server layers
  - Secured channel sinks for .NET remoting.
  - Trusted and encrypted DB communication (ADO.NET data provider).

Client side components

- Windows Forms controls with missing functionality.
- User configurability an internationalization subframework.
  - Interaction with authorization subframework for extended configurability.
- Third party components are required to consist solely of managed (IL) code due to zero deployment requirements.

Service layer components

- ADO.NET abstraction layer (SPoR).
- Abstraction layer for different service component activation models (in-proc, webserver, remoting server, COM+ etc).

Infrastructure components

- Installation utilities.
- Licensing subframework.
- Internationalization helpers and utilities.

Framework Components - Outlook

- Code generation tools in some areas (for example the data access layer).
- Addition of ready-to-use business cases (CIS+).
- Addition of business patterns (workflow patterns etc).

Framework Extensibility

- Extensive use of design patterns, examples:
  - Abstract factory pattern in Data Access Layer components.
  - Adapter class pattern in main window's dynamic menus (menubar, outlook bar, panel bar etc).
  - Observer pattern.
- Aggregation and inheritance.
- Source code available to customers.

Testing provisions

- Framework itself: built-in unit tests (used for regression testing as well).
- Blueprints and samples for application unit testing using NUnit.
- Provisions for automated builds (NAnt).

Considerations for OSS

- Rationale: Provide an alternative, not replacement for .NET.
- Project in cooperation with HSR, cosponsored by the Swiss Federation.
- MONO chosen as platform.

Considerations for OSS (cont'd)

- Divided into subprojects:
  - Service layers support for MONO.
  - Client layer support for MONO.
  - ADO.NET DataSet interoperability for the Java platform.

Service layers and MONO

- Mostly no-brainer, works out of the box.
- Only problematic spot: non-mature nature of WSE implementation.
- (DBMS-specific application code portions might require porting, for example stored procedures).

Client layer and MONO

- Portable UI toolkits/libraries available (Gtk#), but these are not usable due to incompatibility with third party components.
- Windows Forms support in MONO is in pre-alpha state.
- Winelib approach chosen (Windows emulator for ported applications).

Client layer and MONO (cont'd)

- For starters, `System.Windows.Forms.dll` from the .NET redistributable package is used.
  - Surprisingly results are not bad and mainly depending on Wine(lib) maturity.
- Interoperability with OpenOffice?

DataSet for the Java platform

- Rationale: toolkit compatibility for Java clients.
- Project status: basic functionality can be demonstrated.
- Limited to a subset of DataSet's object model required for web service client functionality.

Closed Open Source model ...

- ... and its challenges.
- Customers have read-only access to CVS repository with framework source code.
- Cdot profits from code review and suggestions made by customers.
- Good documentation is required in order to work.

Closed Open Source model (cont'd)

- Unfortunately, this is no true peer-to-peer model.
- Reflections take place to transform parts of the framework to true OSS.

The “Problem” with customers ...

- ... and its solution (fragments of a business model)

The “Problem” with customers ...

- ... and its solution (fragments of a business model)
- Partnership relation to customers, thus calling them “partners”.

The “Problem” with customers ...

- ... and its solution (fragments of a business model)
- Partnership relation to customers, thus calling them “partners”.
- Main objective: provide SPoR also on this level.

The “Problem” with customers ...

- ... and its solution (fragments of a business model)
- Partnership relation to customers, thus calling them “partners”.
- Main objective: provide SPoR also on this level.
- Despite the partnership relationship, customer care tends to be time and resource demanding.

The “Problem” with customers ...

- ... and its solution (fragments of a business model)
- Partnership relation to customers, thus calling them “partners”.
- Main objective: provide SPoR also on this level.
- Despite the partnership relationship, customer care tends to be time and resource demanding.
- Typical “chosen victim” pattern.

Virtual Company and Real People

- Caution: Do not underestimate personal contact with coworkers even in a geographically distributed organization on a regular basis.
- Personal identification with the company is absolutely crucial.

That’s it then

Thank you very much!