

Reduce development costs by up to 80%

Use any XML Industry standard

Modify business rules "on-the-fly" using XEngine

Use 100% XML to assemble complete web solutions

B2B integration and application integration at a fraction of the cost of traditional middleware

Construct applications from reusable business patterns

hyfinity[®]
Intelligent XML

morphyc
architecture

NEXT GENERATION
WEB SERVICES PLATFORM

Morphyc Architecture

Hyfinity[®] has developed a new technology architecture based on XML and Web Services. This architecture is known as Morphyc. The architecture is designed to simplify the complexity associated with the development of distributed business solutions. Morphyc can be used to build systems that are highly flexible and cost efficient. The architecture is founded on native XML technologies that enable its power and flexibility, reducing development costs by up to 80%.

The Morphyc architecture has two core components, XPlatform and XDE. XDE is used to construct applications, which execute on XPlatform.

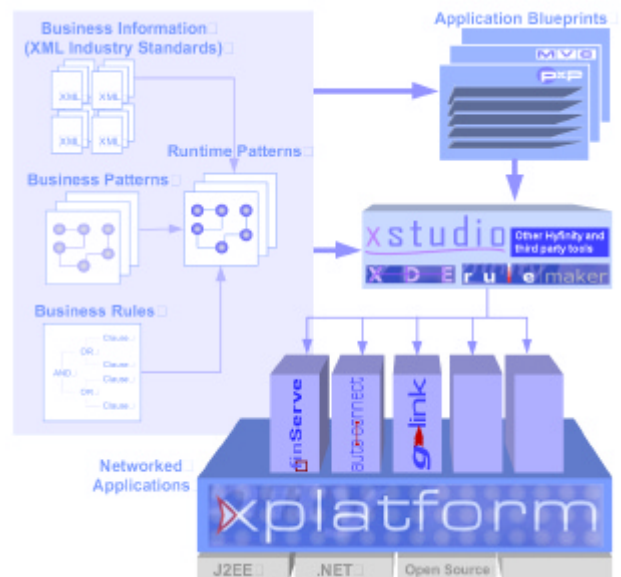
XDE (eXtensible Design Environment)

XDE uses a new approach for developing large distributed solutions. The development process closely resembles physical construction methodologies, such as those used by architects to construct buildings or factory production lines used to build cars.

- **XML Assets** – These are a collection of XML files, that will typically represent industry standards, such as ebXML, XBRL, etc. Using the analogy of the car production line, this is similar to acquiring the raw materials that are used to build the car. For example, doors, chassis, engine, body frame, etc.
- **Patterns** – These are proven and tested patterns for building things. The pattern for the car may be to construct the chassis first, then the engine, following by the body frame, then the doors.
- **Runtime Patterns** – These are clones of the patterns. The key difference is that the runtime pattern is the actual building process, whereas the pattern is just the definition of how it should be done. For the car example, the runtime pattern actually has real assets used to build the car. The pattern may indicate that the car requires four doors, but the runtime pattern will actually use four physical doors with associated part numbers and locations. Runtime patterns use the assets defined above.

- **Blueprints** – These define the way in which something should be constructed. Blueprints use a collection of runtime patterns and create associations between runtime patterns to produce a complete product. In the car analogy, different blueprints will define a 4 door saloon or a 2 door hatchback. It will also define the relationship between different parts of the car and how they associate with each other.

In the real world, it can be argued that the runtime platform for the car is the open road. The car is a physical thing. In software terms, what XDE will



create after this process is a collection of XML documents and other resources such as images, which is the software equivalent of the physical car. The runtime platform for applications constructed in XDE is XPlatform. With Hyfinity's development approach, there is no further requirement for system construction. The application resulting from XDE is directly executable on XPlatform.

A New Approach

Experienced developers as well as architects, designers and business managers can use XDE. Very sophisticated applications can be constructed in a fraction of the time compared to traditional approaches using 3GLs, and the approach still incorporates standard modelling and construction technologies most technologists are familiar with.

The design of Morphyc was influenced by the huge amount of time usually associated with the "plumbing" aspects of distributed solutions. In most projects the smaller proportion of total time and cost is associated with building the actual business logic itself.

The discussion so far has illustrated how the overall framework of a distributed application can be constructed quite quickly by re-using XML Assets and Runtime Patterns, derived from Patterns, and associating these in a manner as defined by a Blueprint. Examples of blueprints include PxP (peer-to-peer web services for connecting business processes within and beyond organisation boundaries) and MVC (XForms-based XML web application development and runtime). Other blueprints are planned for the future and they all follow the same generic architecture.

This section introduces the elements, which perform the actual processing. Every Runtime Pattern consists of interacting Nodes. Nodes communicate by passing XML messages over HTTP (Internet protocol). This enables nodes to be distributed within and beyond organisation boundaries. A Morphyc application can be distributed almost anywhere in any configuration.

- **Nodes** - Nodes interact with each other using XML messages. Within each node these messages are inspected to make decisions based on their content. Based on these decisions other nodes may be invoked. Each node contains logic, in the form of XML rules, known as **Logicsheets**.
- **Logicsheets** – Logicsheets are used to make decisions based on the content of the incoming XML messages. The messages and logicsheets are in XML format and this leads to a declarative system, which can be configured dynamically, without having to compile code every time changes are required.

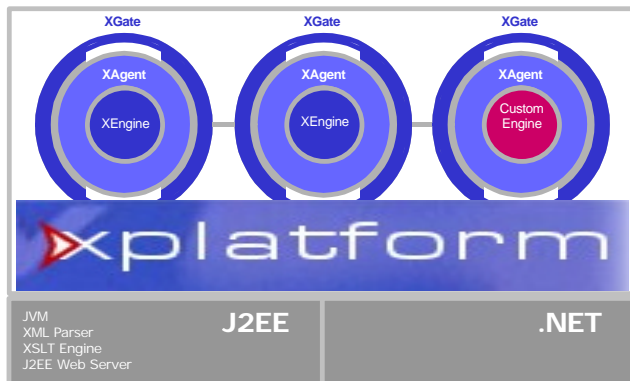
This approach also ensures that in-depth development experience is not required to construct systems. The development approach is

very similar to most case tools. However, one fundamental difference is that the resulting specification, which adheres to a blueprint, is directly executable on **XPlatform**. This eliminates the build phase and saves vast amounts of time and effort for the initial build and also benefits from easier future maintenance. XDE incorporates a graphical editing tool for constructing rules.

Once an application has been designed in XDE, it can be deployed using XDE's deployment features. XDE provides fine-level application distribution controls. For example, every single node can be deployed in a different web environment or the complete application can be deployed on a single laptop.

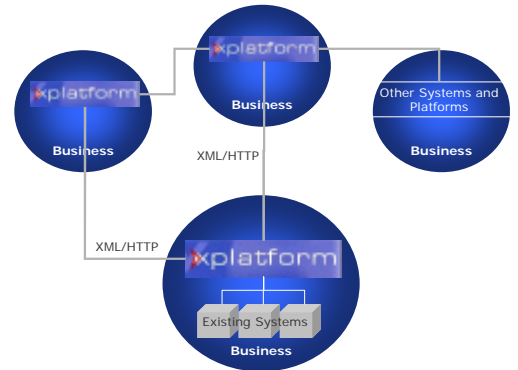
XPlatform

XPlatform provides a runtime platform for executing applications resulting from XDE. XPlatform is constructed using Java and XML and can execute on almost any platform supporting a JVM. XPlatform contains three core components, which together represent the runtime version of a node:



- **XEngine** – XEngine is a forward-chaining rules engine used to process incoming and outgoing XML messages using logicsheets. For situations where a native XML solution is insufficient, *custom engines* can be developed using Java classes and used within XPlatform
- **XAgent** – XAgent provides an "incubator" for XEngine. XAgents collaborate by exchanging XML messages. Agents can be deployed anywhere and discover each other dynamically at runtime. XAgents can be deployed locally within a JVM as native Java services, or they can be deployed as web services using XGate.

- **XGate** – XGate is a generic gateway, providing access over different protocols, including HTTP and Java. XAgent uses XGate to locate other XAgents, which are web services. XGate is also used by external web services to locate local XAgents



Advantages

The **Morphyc architecture** has the following advantages:

- **Rapid construction** of solutions using pre-defined patterns, saving time and cost
- **Dynamic business rules** that are executed by a sophisticated rules engine. This enables the creation and modification of business rules "on-the-fly", without incurring massive re-development costs
- **Solutions are based on XML industry standards.** Most of these standards are free and can be readily re-used. This saves a significant amount of analysis and development time, saving costs
- **Solutions are tailored to customer needs** without obsolete functionality. This is better matched compared to packaged products and provides all the flexibility of a bespoke solution – "best-of-both-worlds"
- **Business Oriented development.** The Morphyc architecture removes the technical complexities associated with large distributed solutions and enables focus on the business problem
- **No legacy foundations.** Constructed "from the ground up" using XML and Web Services technology
- **Can be used to construct B2B, B2C and system-to-system solutions** in weeks rather than months or years