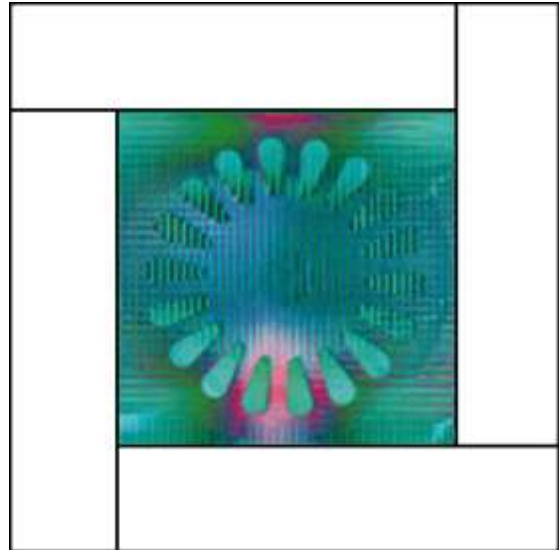


# AUTHORIZE



## **Authorisation - An Aspect Oriented Service**

**Andrew Meehan  
Forge Research Pty Ltd**

Forge  
Suite 116, Bay 9  
Locomotive Workshop  
Australian Technology Park  
Cornwallis Street, Eveleigh  
NSW 1430 Australia

Phone: +61 2 9209 4152  
Fax: +61 2 9209 4172  
Email: [info@forge.com.au](mailto:info@forge.com.au)  
Web: [www.forge.com.au](http://www.forge.com.au)

**FORGE**



# Table of Contents

<b>Abstract.....</b>	<b>2</b>
A Changing Enterprise Landscape.....	3
Easing the Pain.....	3
Aspect Oriented Services.....	4
Authorisation as an Aspect Oriented Service.....	4
Aspect Oriented Authorisation Service Essentials.....	5
An enterprise authorisation control process.....	6
Benefits of Aspect Oriented Authorisation.....	8
Analysis of Current Authorisation Technologies.....	8
<b>Conclusion.....</b>	<b>9</b>
<b>Glossary.....</b>	<b>10</b>

## **Abstract**

In an environment where real benefits are being seen from the centralised control of some critical business functions, Authorisation has become the poor cousin as companies start 'single sign on' type of initiatives for security control. The realisation that Authentication is only one facet of the security framework which needs a flexible, centralised, technology independent Authorisation mechanism to provide real corporate access control is beginning to dawn in the minds of forward thinking corporate IT strategists. This paper introduces the concept of and need for aspect oriented services and examines enterprise Authorisation from this perspective.

## A Changing Enterprise Landscape

The Enterprise IT landscape is changing. The demands for cohesive customer service requires new integration between systems that used to behave as islands. More stringent reporting requirements mean that more metrics must be gathered and aggregated across many systems. The quest for ever greater cost savings and efficiency is mandating that more and more operations procedures be automated. And then there's the Web! All of this leads to scenarios where access to systems by users or other systems is on the increase. This access is also more specialised than ever before. Users are no longer trusted employees and it is important that what users are allowed to do across the myriad of systems they access, either directly or indirectly, is more controlled than ever.

Coupled with this is a new emphasis on security. Companies are looking to better authentication and 'single sign on' type solutions to get more control over who is accessing which systems. This is a good first step but it is important for companies to understand the distinction between Authentication and Authorisation. No matter whether digital certificates or name and password are used to identify users, authentication is still the process of determining that the users, systems or devices are who they say they are. Authorisation is a post authentication process that determines what users are allowed to do. This goes beyond whether or not they have access to a particular system, it covers what functionality is available after they have been given access to a system.

Many authorisation systems allow access to be controlled at a system function level but more and more there is a need for greater control at both a macro and micro level. Examples of this at the micro level could be specifying that a user or system is not just capable of executing a command, but that only certain values for that command are allowed for that user. At the macro level, an example would be to limit system access or access to particular functions within systems, based on whether a user was physically in the office or using remote access. Another macro example would be one where data input personnel were only allowed to input data up to 5pm, but their supervisors could still have reporting access until 8pm.

Today's enterprise environment is in a state of flux. A typical enterprise has a myriad of different technologies. Even if you stay with a single vendor, technologies change over time. For Sun Microsystems this could be systems written in C/C++/Java (J2SE) moving to J2EE, EJB and Web services. For Microsoft it could be VB, Visual C++ moving to .Net and web services. Many companies have a mixture of these and other technologies. As companies contemplate their strategic technical direction, they are being pressured into selecting single technology streams for all new applications. History however shows that even if this is a good idea (and one size does not always fit all), technologies and vendor offerings change over time. Consequently it is becoming increasingly difficult for an enterprise to obtain a consolidated view of the authorisation policies in effect across its many and diverse systems or to enforce centralised control over who has access to what.

The result of all this is tension. Developers want freedom to use best of breed solutions. IT strategists want a common platform, reusability and secure corporate-wide access policies. The business wants everything connected and accessible from new access points such as the web or B2B web services. Unfortunately these are valid tensions. If inappropriate technology is used, then projects may fail. If there is no technology plan then anarchy results. If there is inadequate security then the entire corporation could be at risk. And all these are incidental if the business cannot get the required technology support to remain competitive.

## Easing the Pain

While there is no silver bullet to ease this tension, there are some steps that can be taken to ease the pain.

1. Think of process as more important than technology.
2. Try to determine the likely integration points required between existing and proposed systems, the functionality required, and then create an integration plan based on suitable middleware standards.
3. Analyse proposed system development requirements and create technology guidelines that propose an approved range of technologies that are suitable to meet those requirements. Create repositories of reusable components that are published in library format for particular technologies and in specification format for cross-technology reuse.

4. Provide a process where projects can use technology outside those guidelines which ensures that a) a case is made illustrating why the standard technologies are unsuitable and b) the project can still meet its integration obligations with respect to other enterprise systems.
5. Determine which services should be governed at the corporate level and abstract responsibility for these away from individual applications and development teams.

## Aspect Oriented Services

Let us focus on this last point and introduce the concept of aspect oriented enterprise level services. These are based on the practice of aspect oriented programming. Aspect oriented programming is a relatively new programming method. It is based on the premise that within an application, programmers should concentrate on the business logic and have 'concerns' such as logging or authentication/authorisation layered over the top by another means. The main driver for this is that

1. it allows programmers to focus on the business function without having to concern themselves with subsidiary functions
2. it allows those subsidiary functions (concerns) to be maintained/upgraded separately from the business logic
3. it enables standardised and controlled implementation of these functions across all parts of the application.

Aspect oriented services take this concept to the next level. Instead of separating these concerns within an application, certain service functions are identified as those that should be controlled outside of the application itself. Authentication, Authorisation, Auditing and Billing are candidates for this type of service. The advantage of this approach is that corporate-wide policies can be put in place and enforced across many applications to facilitate the corporate view being mapped onto the application system infrastructure. Further to this, once these services are in place, modification of these policies can be done at a central place with no changes to the underlying applications required.

## Authorisation as an Aspect Oriented Service

Let us examine Authorisation as an aspect oriented service. In real life, system users be they staff, customers or other systems have one or more roles that they play within an organisation. A user's interactions with these roles may vary constantly as they get new responsibilities or lose old ones. They may temporarily stand in for their manager or may be seconded two days a week to another role. Their level of system access may also be controlled by their physical location and/or the time of access. For example, they might be allowed full control over some printers but only have limited access to others.

In short, enterprise level authorisation presents a landscape that is more to do with people than systems. Many authorisation decisions are based on company policies and work procedures and cannot be foreseen at the time an application is written let alone be provided for within that application. The authorisation policies govern groups of systems, roles, users and commands like a general commands an army. Each application can only perform a part of the entire ensemble. The policies as a whole must be flexible and cohesive enough to enable corporate-wide changes to be defined and enforced with ease.

Let us take our earlier data entry scenario a little further. In a data entry department we may have a manager, several supervisors and a number of trained data entry clerks. The manager may have access to reporting facilities provided by the application. These could be as simple as how many transactions a particular clerk has performed, or the rate of errors for the whole department. The supervisors could also act as data entry clerks, but with the extra power to correct errors on previously completed transactions. The clerks may access a number of systems to get input information, verify that information is acceptable (e.g. a payment was received) and then enter a work order into the provisioning system. All is well.

Soon everything gets better. Business is booming. Orders are streaming in. But something is wrong! The work order entry is falling behind. There is not enough time in the day nor trained data entry clerks to keep up. Management decides to bring on an extra shift. In order to staff this shift, a number of casual untrained clerks are taken on. The trained staff shall split into two groups with each half staffing a shift accompanied by half the

casual staff. Given their lack of training, casual staff shall only be allowed perform work orders for non critical jobs with the rest being handled by the experienced staff. New authorisation policies shall be created that

- a) allow one of two time brackets for access to be assigned to each clerk or supervisor.
- b) define a new role for casual staff who do not have access to credit or payment information for critical customers.
- c) confine work order entry for casual staff to non critical jobs.

The work order entry catches up and everybody is happy again. The company has now bought time to automate the entire procedure based on yet another set of authorisation policies.

While this example might be fanciful, it does help to illustrate the difference between application level authorisation and an aspect oriented authorisation service. Using application level authorisation each of the affected applications may have had to be recoded to add the new roles, and the new time and content constraints. This could have taken just enough time to kill the plan if not the organisation.

In an aspect oriented environment, a new role would have been created and associated with all of the systems in question. The casual workers would be created and added to that role and any new constraints captured. The new time constraints would be added for all roles. All the systems would then be in sync and the plan can carry on. When the automated procedures are implemented, the old systems or new adapters are given roles and new policies for them are engaged. The casual role is discontinued and the data entry clerks are moved to new roles within the organisation. The process where live applications are changed is avoided as much as possible.

The above example is feasible because there is one unified view of what each user is allowed to do and there is one mechanism that needs to be learnt to enable a user to be plugged into systems across the enterprise.

## **Aspect Oriented Authorisation Service Essentials**

1. It must support the entities to model real world scenarios – users, roles, systems, commands, targets, etc.
2. It must be technology independent to allow control over many systems across the enterprise.
3. It must handle very different scenarios across many technologies concurrently.
4. It must handle application independent requirements such as time, location, dollar amount or other multiple dependencies.
5. There must be a unified (although specialised consoles for different areas could exist) mechanism for the creation and lifecycle control of policies across the enterprise. That is, it should provide an abstraction layer between the policy language and the environment to which it applies.
6. One or more consoles can control as much of the enterprise authorisation requirements as required using one or more centralised repositories of authorisation rules.
7. There must be a mechanism for clear and unambiguous resolution of conflicting policies.
8. Meaning of commands, targets, actions etc must not be constrained so that real life systems can be modelled and not just file or systems access.
9. It must be capable of scaling up to meet enterprise demands as more systems are added to the federation.
10. It must provide the authorisation equivalent of single sign on. Once the user has been identified, a map of their access capabilities controls their every move through the enterprise.

11. Policies must be capable of being constructed from pre-built policies. This is important for large organisations. For example, a system administrator could be made up of a general user role augmented by administrator access rules or there may be a set of cascading roles that provide greater details of access from general employee to job role within department.
12. One policy can cover many resources. This helps avoid inconsistent policies on different resources.

## **An Enterprise Authorisation Control Process**

The difficulties involved in creating an enterprise wide authorisation framework controlling systems as diverse as HR, web-services and building-access cannot be underestimated. Two possible mechanisms to achieve this are strategic 'top down' or disparate 'bottom up' approaches.

The top down process is the strategic approach. Time is needed to get management buy in, to find the right resources and to create the right models and policies. The chances of not getting going at all are greater but the chances of long term success are greatly increased with proper planning and management sponsorship. This approach involves the creation of an enterprise owner for authorisation initiatives. The CTO is a candidate but more and more roles such as the Chief Security Officer (CSO) and the Chief Privacy Officer (CPO) are emerging to fulfill this requirement. The CSO and his/her staff, complemented by appropriate systems and business resources, build a high level model for the authorisation requirements of the company. This leads to a system level security contract for each participating system. The security contract defines the systems responsibilities with regard to the company policies and the authorisation infrastructure obligations with regard to the systems. In turn this becomes part of the specification for individual systems. The collection of security contracts also forms the basis for scoping the necessary authorisation infrastructure and for compliance audits. Such an approach provides real support for demonstrating IT Governance based on a coordinated plan and the ability to produce consolidated reports and risk assessment.

The bottom up and top down approaches are very similar. A bottom up approach results in authorisation across the enterprise becoming a collection of system or departmental islands that join together over time without any real sponsorship from senior management. One of the problems with this approach is that the IT Governance benefits of the authorisation service are largely ignored. Also the chance that the islands might use incompatible technology is greatly increased. The seeming benefit of this approach is that there is less planning and cost involved to get going although it is doubtful that this benefit is sustainable. However, one advantage of this approach is that if a single authorisation mechanism is approved for use enterprise-wide, then disparate systems can implement this authorisation scheme with the knowledge that as an enterprise-wide authorisation rollout is implemented, these systems are pre-configured to use the new system after their specific policies have been folded into the larger scheme.

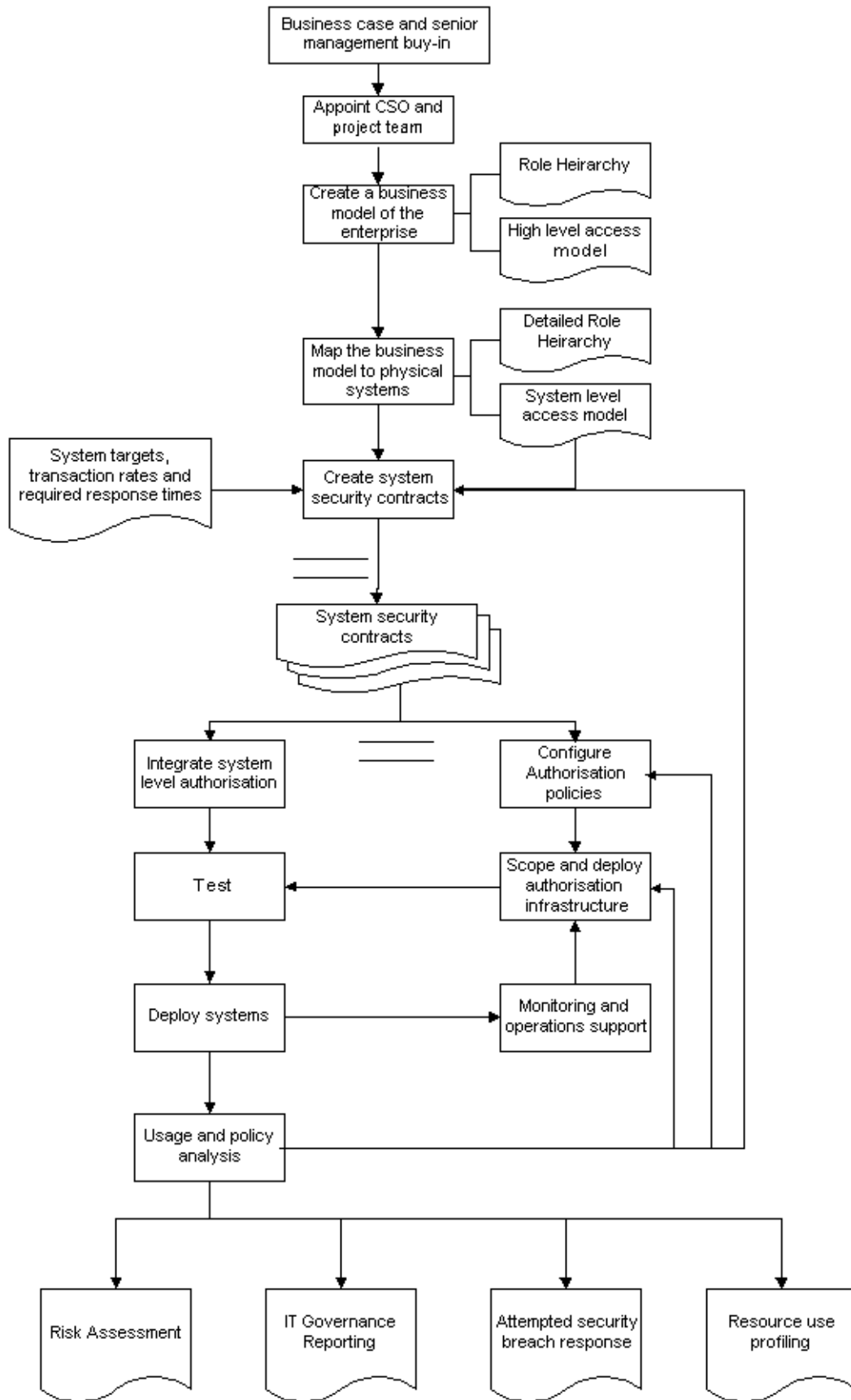


Figure 1. A top down process flow

## Benefits of Aspect Oriented Authorisation

Most of the benefits are similar to the requirements.

1. The company can implement centralised control points for sets of systems or for the entire enterprise. This is independent of the technologies used by those systems. This means that corporations have greater control over who has access to what systems.
2. System/Policy Administrators save time and money by deploying Authorisation policies once using a single mechanism, as opposed to having many disparate policies across many systems using different authorisation mechanisms.
3. Developers save time and money. They use existing components which plug into their applications to get enterprise approved authorisation functionality.
4. Policies can cover constraints imposed outside of the application domain such as time or physical location, to ensure that they are in sync with the real life requirements of the company.
5. Users can be given immediate access to dozens of systems simply by adding them to one or more predefined roles. Similarly access can be removed, or the access required by changing jobs, given in one centralised step.
6. Consolidated Enterprise wide Authorisation access logs provide real business benefits from the perspective of IT Governance
  - Analyse successful and unsuccessful requests across all systems to see who did what and when.
  - Detect suspicious patterns of activity.
  - See which systems honour their security contract or make inadequate use of authorisation.
  - Analyse enterprise resource load, availability and usage patterns.
  - Understand the authorisation model implemented for the enterprise.

## Analysis of Current Authorisation Technologies

In general, today's systems provide authorisation in one of a number of ways.

- Application-specific authorisation rules, which control individual systems with various degrees of granularity or flexibility. These are not aspect oriented and the application has to be reconfigured or even recoded to change the authorisation rules.
- Application server level authorisation.  
This may be aspect oriented to a point such as JAAS or .Net, but the rules are generally stored in flat files and are defined on a per server basis. These types of systems mandate a particular implementation technology and there is no clear corporate level control.
- Aspect oriented authorisation.  
These systems provide true application and technology independent authorisation services at the enterprise level. Candidates that fall into this category are the Oasis XACML initiative, which is still in an early form and the Forge Authorization Service, which has connectors for a number of technologies including EJB, CORBA and web services.

## Conclusion

In a similar fashion to the benefits that accrue from organisations having controlled and standardised desktops, centralised control over authorisation policy deployment and enforcement can offer real security and cost benefits to an organisation. The use of aspect oriented services also helps development projects come in faster and cheaper as concerns are separated from the application logic. All application or technology specific authorisation schemes have limitations that can be addressed by aspect oriented authorisation services. Information Technology strategists need to weigh up the consequences of continuing down the individual application approach against the future benefits of pursuing the aspect oriented way.

## Glossary

Term	Description
C++	The C++ (C plus plus) programming language
Java	The Sun Microsystems Java platform
JAAS	Java TM Authentication and Authorization Service
XACML	XACML is an XML-based access control policy language: a standard way to say who can do what, when.
.Net	Microsoft .Net programming environment
EJB	Enterprise Java Beans
J2EE	Java 2 Platform, Enterprise Edition
Policy	A high-level overall plan embracing the general goals and acceptable procedures in force within the enterprise. In Authorize, a Policy ultimately translates into a set of Rules, the evaluation of which yields an ALLOW or DENY authorisation action.
J2SE	Java 2 Platform, Standard Edition