Modern distributed computing systems, including cloud based ones, require much more than simple access control lists. Policy based systems provide this much needed scalability through the enforcement of rules, most flexibly written in terms of the attributes of the subject, resource, action and environment (or context). Attribute based access controls (ABAC) are a generalisation of the well-known role based access control model (RBAC), in which the access control rules are based on any attributes of the subject, and not just his or her role in the organisation. However, fixed policy rules based on the static attributes of the system entities are not flexible or dynamic enough to cater for all of today’s requirements. We need to be able to intelligently and dynamically update the access control rules as well as the attributes of the various system entities. In this talk I will describe some of the research we have carried out at the University of Kent that does just this.

Distributed systems often have policies that apply collectively to all the resources of the system. A common example is the bank ATM network that controls the amount of money a person can withdraw per day from all ATMs in the network. Simple access control lists on each resource cannot enforce such policies. Neither can policy based ABAC systems that do not keep a historical record of previous transactions (i.e. state information). Unfortunately the majority of today’s policy based access control systems are stateless. We will describe a novel use of policy obligations that allow policy rules in stateless policy decision points (PDPs) to introduce dynamic state into the access control system, in order to dynamically enforce policies throughout a distributed system.

Subjects who provide their personal information to one system or organisation, may wish to control its dissemination throughout a distributed system. We will describe the design and implementation of a sticky policy access control system in which the subject’s privacy policy is stuck to his or her personal data, and stays with it regardless of which computer system the data is subsequently transferred to. The access control system itself is responsible for storing, retrieving and enforcing these sticky policies, so that all the applications needs to do is recombine the data and sticky policy before transferring the personal data to another system.

Sometimes computing systems, and the humans who program them, are not intelligent enough to cater for every possible circumstance that may arise. Access control systems that are similarly constrained will either deny access to someone who should have been granted it, or grant access to someone who should have been denied it. In hospital environments, it could be a matter of life or death if a medical professional is wrongly denied access to medical data in an emergency. We will describe a Break The Glass access control system that relies on the intelligence of the subject, to decide when he or she has been wrongly denied access to a resource, and that allows them to override the deny and gain access. Frivolous or malicious breaking of the glass to gain wrongful access is prevented through the access control system notifying management when the glass has been broken, so that the subject is required to explain his or her reasons when the emergency has passed.

No matter how clever the security administrator is, he or she may get some access control rules wrong. We will describe our research into the autonomic control of federated ABAC systems that monitor the underlying policy based access control system to ensure that it is working according to the higher level goals of the organisation. If not, the autonomic system will dynamically modify both the access control policy rules and the attributes of the subjects in order to bring the ABAC system back in line with the organisation’s intended goals.