Contracts abstract away detailed implementation information of individual actors and model dependencies between them. 

Contracts can be linked to more general social structures and can be deployed and, therefore, allow software engineers to perform a more intuitive analysis.

Contracts are publicly observable, improving run-time monitoring of the system as a whole. Contracts correspond to the types of relationships occurring in organisations in which business software systems are indeed applications themselves. Some efforts have sought to use methodologies for agent oriented software engineering (e.g., [2]), such as the conference management system in the case of the conference management system, the case of [2], but these are to some extent toy examples rather than real use cases.

While several efforts have been made to report on particular applications that are inadequate to show broad applicability and relevance of generic techniques across a range of domains or methodologies for agent oriented software engineering (e.g., [2]), these have tended to be one-off results from real business cases, and developing prototype systems for unashamedly tied directly to our efforts in eliciting requirements may also highlight open questions for research into fundamental barriers to the adoption of agent technologies, the architecture, the project also seeks to address one of the open agent systems.

Contracts can model functional as well as non-functional requirements. In contrast, our effort to develop a contract architecture is more than focus on simply building particular instantiations of an open agent systems.

The CONTRACT project [11] aims to develop and apply electronic contracting and contract-based monitoring and verification techniques in real world applications. However, rather than focus on simply building particular instantiations of an open agent systems, our project aims to develop and apply electronic contracting and contract-based monitoring and verification techniques in real world applications. This paper presents results from the initial phase of the project, which focused on requirements solicitation and analysis. Specifically, we survey techniques in real world applications. This paper presents results from the initial phase of the project, which focused on requirements solicitation and analysis. Specifically, we survey techniques in real world applications. However, rather than focus on simply building particular instantiations of an open agent systems, our project aims to develop and apply electronic contracting and contract-based monitoring and verification techniques in real world applications. However, rather than focus on simply building particular instantiations of an open agent systems, our project aims to develop and apply electronic contracting and contract-based monitoring and verification techniques in real world applications. However, rather than focus on simply building particular instantiations of an open agent systems, our project aims to develop and apply electronic contracting and contract-based monitoring and verification techniques in real world applications. However, rather than focus on simply building particular instantiations of an open agent systems, our project aims to develop and apply electronic contracting and contract-based monitoring and verification techniques in real world applications.
Limited in Ireland and other countries.

Trade mark of The European Computer Driving Licence Foundation certification business domain (see Figure 1 for an overview):

There are three categories of business entities cooperating in the certification programmes of similar structure.

Later in Slovakia, and can be equally well applied to other application area, business entities and services involved in the provision of certification services. The model has been heterogeneous and independent businesses to flexibly collaborate on the implementation of the contracting infrastructure. This is currently demonstrated the business benefit to be gained. Note that it is not commercial context for the development of work in this area, and therefore, the whole process is very costly. Nowadays, the insurance domain relies heavily on traditional ways of claims handling. Every aspect of a claim is dealt with by different specialists working in different departments of a company, or in different companies involved in the total claims handling process.

Thus the insurance market is increasingly seeking ways to economise on management and monitoring should result in significant reduction in labour cost, better utilisation of resources, higher reliability and consequently improved quality of service to the consumer.

2.1.1

Modular certification testing allows a large number of contractors on the supplier market is the primary focus of the use case.

The application of electronic contract-based technology should primarily be applied to enable the dynamic operation of the certification programme; accreditation institutions, test centres, test room operators and accredited testers, as well as test centre can provide all these services internally, in many cases it is advantageous to procure these services on the supplier market. Traditionally, test centres to establish and manage supplier relationships with accredited test room operators and accredited testers, as well as to supervise/test marking services, respectively.

• Test centre
• Candidate testing is a business-to-consumer business relationship.
• Certification test contract is a business-to-consumer relationship.
• Test centre can provide all these services internally, in many cases it is advantageous to procure these services on the supplier market.

2.1.2

CarRepairGrid (Figure 2)

Test Room Rental

Tester Hire

Certification Test Contract

• DamageSecure (Figure 2)
• The Dynamic Insurance Settlement use case describes a use case showing business entities and services they provide
• Witnesses, surveyors/experts, lawyers, insurance companies, claims handling by increasing the level of process automation and improving the integration of the different parties (e.g., victims, doctors) and systems involved.

2.2

Use Case 2: Dynamic Insurance Settlement

Out of a number of different contracts governing the provision of services in the domain, the use case primarily focuses on the following:

- Test Room Rental
- Tester Hire
- Certification Test Contract

Contracts

Benefits of Contract-based Technology

Contracts

Automation of contract execution and management should result in significant reduction in labour cost, better utilisation of resources, higher reliability and consequently improved quality of service to the consumer.

The emphasis was on capturing business relationships and contracts in each domain as these provide a solid foundation for the description of technical requirements and solid labels provided.

The use cases were captured in a multi-round dialogue, the basis for the recording of the business model, and selected main contracts formally reported in a separate paper [12].

Each industrial partner: modular certification testing (provided by Certicon); service procurement in the insurance industry (Y’All);

Requirements in four application areas, each represented by one specific use case scenarios. In this paper, we provide only a brief overview of each of the four use cases – for detailed descriptions, see [10].
all businesses involved in dealing with car damage claims for several insurance companies. The goal of DamageSecure is to enhance the quality and efficiency of the total damage claims handling process between consumers, damage repair companies and insurance companies. CarRepairGrid reasons about repairs of damaged cars that are insured at insurance companies in order to settle the claim under the best circumstances (lowest prices, highest quality, as soon as possible, as close as possible, etc).

2.2.1 Actors
In all, five types of entities are involved in business transactions in the insurance domain:

- Customers who are the holders of insurance policies
- Repair companies which repair and replace damage.
- Insurance companies which inspect, approve and pay approved claims.
- DamageSecure which operates the CarRepairGrid and acts as a broker between the other parties in the domain. It offers services to both Insurance Companies (centralised procurement of services) and Repair Companies (centralised selling of services to Insurance Companies).
- Experts who perform counter-expertise for Damage Secure

2.2.2 Contracts
In terms of contracts regulating business transactions in the domain, the use case is focused on:

- Overall Contract which specifies the relationship between an Insurance Company and DamageSecure.
- Repair Contract which specifies the relationship between DamageSecure and a Repair Company.

The Overall Contract is a good example of a long-term contract between a supplier and a broker defining a contractual framework within which targeted, short-term repair contracts are established. Repair Contracts are the primary interest from the project's perspective because they have significantly higher volume and dynamics than the rather static Overall Contracts.

2.2.3 Benefits of Contract-based Technology
The biggest opportunity for the application of automated contract technology lies with DamageSecure. Contract technology can significantly improve insurance claims handling by enabling automated, contract-based matching of repair requests to repair companies as well as automated monitoring of the claims handling process. This is expected to lead to a decreased cost due to reduced manual labour, increased competition and improved efficiency of the claims handling market. With an estimated 100,000 claims per year, the automation of CarRepairGrid could potentially save 172,000,000 Euros. Greater variety of customised insurance policies and a wider range of repair options together with accelerated claim handling are the other benefits expected from automated contract-based claim handling.

2.3 Use Case 3: Aerospace Aftermarket
The aerospace aftermarket is increasingly populated by customers buying a service rather than a product. Here, the aircraft engine manufacturer is responsible for providing a specified number of serviceable engines so that the airline operator's aircraft can be kept flying. The engine manufacturer is paid by the hour when the engines are available and may face a penalty if planes are on the ground waiting for a serviceable engine. In this business model, servicing and maintenance becomes a key driver of long term profitability for the engine manufacturer. Aftercare contracts are worth millions of Euros and can last several years. They are complex with stipulated service levels and penalties for failure to meet them.

A unique feature of this use case is the Aerogility system, an agent-based decision support tool developed by LostWax to simulate aerospace aftercare. A contract-enhanced Aerogility would be able to show the effects of variations in contracts – not only in profitability but also the different aftercare strategies needed to meet the revised contract.

2.3.1 Actors
From the perspective of the use case, there are three relevant types of businesses in the domain (see Figure 3):

- Airline Operators are customers for aftercare contracts. Each operator has its own fleet of aircraft which need to be kept in service.
- Engine Manufacturers are suppliers of aftercare contracts. They attempt to fulfil the service levels specified in the contracts or else incur penalties.
Service level agreements (SLA) play an increasing role in IT.

2.4

fulfilled.

recommendations of change on the basis of current contracts to be

operators and manufacturers can investigate the properties of
domain. By using contract monitoring and verification techniques,
emerging from contract-driven interaction between parties in the
emphasis of the use case is on investigating collaboration patterns
than automating operation in the application domain, the
environment provided by the Aerogility simulation system. Rather
here the contracting technology will be used in the simulated
of the developed technology directly into the application domain,
In contrast to the other use cases, which envision the deployment
developed contract-based technology.

frequently created and may provide a useful extra test of the
Aftercare Contract, but is potentially more dynamically and
Supply Contract is more speculative and simpler than the
large number of specific service requests are handled. The Parts
relatively long-term, they provide a framework under which a
Note that although Aftercare and Parts Supply Contracts can be
•
•
•
following two contracts:

between respective parties. The use case explicitly models the
locations where part supplies should be delivered, the cost of
parts, delivery times, etc.

given period. The Parts Supply Contract specifies e.g.
new parts or refurbished old parts of a given type over a
manufacturer asks a part manufacturer to make and deliver
Parts Supply Contract

Aftercare Contract specifies e.g. serviceable engine rate,
and maintain engines for the operator's aircraft. The
under which the aircraft manufacturer undertakes to supply
given types to engine manufacturers Part Manufacturers

Benefits of Contract-based Technology

Due to compliance issues and legal regulations, IT service
agreement.

main agreement, while they can contain extensions as long as they
for single change requests, which have to meet the rules of the
conditions and obligations, and any number of sub-contracts, e.g.,
also envisioned. A more complicated scenario can involve a
In addition to a simple scenario consisting of just one customer
and one provider, more complicated collaboration patterns are

•
•
•
two key contracts:
The provision of services in the use case domain is governed by
2.4.2

party.

level view of internal services and resources involved at each
the provision of software engineering services, including a high-
Figure 4 depicts business collaboration patterns corresponding to
use case focuses primarily on software engineering services for
development of a new software system. From this broad range, the
provided range from processing a single help desk call to the
deliver the requested services to the customer. IT services
contribute to its business, and an IT service provider who can
deliver technical support processes. Automated contract technology
of software products and the performance of software engineering
providers and customers increasingly tend to measure the quality

•

•

late or with low quality, reduces costs of penalties and increases
violations. This decreases the risk of delivering IT services too
early warnings in case of the risk of not meeting the conditions
performance indicators before and during service delivery, issuing

and reliable operation of the IT infrastructure of an organisation
defined, measured, and continuously improved. To ensure stable
users and offering respective services. In this process, the quality
meeting customer requirements. One of the major tasks of service
providers high quality and flexible services at reasonable cost,

Business collaboration in the domain takes place between a
 Actors

with a high degree of performance, the responsible managers
The ability to store and manipulate contracts is crucial for any contract-based system. It represents the basis on which the more advanced functionality of contract management and monitoring can be implemented. The first group contains generic requirements which concern storage and representation, life-cycle management, monitoring, and verification functionality. Please refer to [10] and [11] for more detailed analysis. In the CONTRACT project, requirements for design-time verification of established contracts have been defined.

Table 1: General requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>The system allows for short-term contracts created only for the purpose of providing an accurate and up-to-date view of contracts and the optimisation of the provisioning process. In scenarios with multiple dependent contracts, requesting changes to already established contracts can be better controlled, avoiding inconsistencies and unnecessary business disputes. In addition, existing contracts can be extended or renewed, potentially with modified obligations.</td>
</tr>
<tr>
<td>R2</td>
<td>A party can have a number of different contracts with different parties active at the same time. The system can detect whether a particular clause in being fulfilled or not. The system can detect violations of active contracts and issue warnings when there is a risk of contract violation. The system can provide information on the fulfilment state of contracts. The ability to evaluate at runtime to what extent the behaviours of the contract are fulfilled can be calculated from partial metrics. Aggregated degree of contract fulfilment can be determined from performance, cost, and service-related metrics. Fulfilment metrics can either be evaluated automatically or entered manually by a human expert. Aggregated degree of contract fulfilment can be calculated from partial metrics.</td>
</tr>
<tr>
<td>R3</td>
<td>The system supports contract templates and operations with them (storage, retrieval, update, and deletion). Contract hierarchies (such as a master contract and all its subcontracts) are supported for contracts and contract templates. Contract instances can be created by filling in details (such as price, or delivery date) into pre-specified contract templates. System allows for obligations that come into force on being triggered by (possibly unpredictable) domain events. The system can detect whether a particular clause is being fulfilled. The system allows for contracts that operate over long, defined periods (as opposed to short-lived one-off requests). The system provides information on the fulfilment state of contracts. The system can detect whether a particular clause in being fulfilled. The system can determine the status which denotes how the contract should be currently treated. During their lifetime, contracts can go through a number of different stages. At each stage, each contract is ascribed a contract status which denotes how the contract should be currently treated. During their lifetime, contracts can go through a number of different stages. At each stage, each contract is ascribed a contract status which denotes how the contract should be currently treated.</td>
</tr>
<tr>
<td>R4</td>
<td>The system allows for long-term contracts created only for the purpose of providing an accurate and up-to-date view of contracts and the optimisation of the provisioning process. In scenarios with multiple dependent contracts, requesting changes to already established contracts can be better controlled, avoiding inconsistencies and unnecessary business disputes. In addition, existing contracts can be extended or renewed, potentially with modified obligations. The system can detect whether a particular clause in being fulfilled or not. The system can detect violations of active contracts and issue warnings when there is a risk of contract violation. The system can provide information on the fulfilment state of contracts. The ability to evaluate at runtime to what extent the behaviours of the contract are fulfilled can be calculated from partial metrics. Aggregated degree of contract fulfilment can be determined from performance, cost, and service-related metrics. Fulfilment metrics can either be evaluated automatically or entered manually by a human expert. Aggregated degree of contract fulfilment can be calculated from partial metrics.</td>
</tr>
<tr>
<td>R5</td>
<td>Representation of IT service agreements in a machine understandable format is required. The system can issue warnings when there is a risk of contract violation. The system detects and reports violations of active contracts. The system provides information on the fulfilment state of contracts. The system can detect whether a particular clause in being fulfilled. The system can determine the status which denotes how the contract should be currently treated. During their lifetime, contracts can go through a number of different stages. At each stage, each contract is ascribed a contract status which denotes how the contract should be currently treated. During their lifetime, contracts can go through a number of different stages. At each stage, each contract is ascribed a contract status which denotes how the contract should be currently treated.</td>
</tr>
<tr>
<td>R6</td>
<td>The system supports contract templates and operations with them (storage, retrieval, update, and deletion). Contract hierarchies (such as a master contract and all its subcontracts) are supported for contracts and contract templates. Contract instances can be created by filling in details (such as price, or delivery date) into pre-specified contract templates. System allows for obligations that come into force on being triggered by (possibly unpredictable) domain events. The system can detect whether a particular clause is being fulfilled. The system allows for contracts that operate over long, defined periods (as opposed to short-lived one-off requests). The system provides information on the fulfilment state of contracts. The ability to evaluate at runtime to what extent the behaviours of the contract are fulfilled can be calculated from partial metrics. Aggregated degree of contract fulfilment can be determined from performance, cost, and service-related metrics. Fulfilment metrics can either be evaluated automatically or entered manually by a human expert. Aggregated degree of contract fulfilment can be calculated from partial metrics.</td>
</tr>
<tr>
<td>R7</td>
<td>The system supports contract templates and operations with them (storage, retrieval, update, and deletion). Contract hierarchies (such as a master contract and all its subcontracts) are supported for contracts and contract templates. Contract instances can be created by filling in details (such as price, or delivery date) into pre-specified contract templates. System allows for obligations that come into force on being triggered by (possibly unpredictable) domain events. The system can detect whether a particular clause is being fulfilled. The system allows for contracts that operate over long, defined periods (as opposed to short-lived one-off requests). The system provides information on the fulfilment state of contracts. The ability to evaluate at runtime to what extent the behaviours of the contract are fulfilled can be calculated from partial metrics. Aggregated degree of contract fulfilment can be determined from performance, cost, and service-related metrics. Fulfilment metrics can either be evaluated automatically or entered manually by a human expert. Aggregated degree of contract fulfilment can be calculated from partial metrics.</td>
</tr>
<tr>
<td>R8</td>
<td>The system supports contract templates and operations with them (storage, retrieval, update, and deletion). Contract hierarchies (such as a master contract and all its subcontracts) are supported for contracts and contract templates. Contract instances can be created by filling in details (such as price, or delivery date) into pre-specified contract templates. System allows for obligations that come into force on being triggered by (possibly unpredictable) domain events. The system can detect whether a particular clause is being fulfilled. The system allows for contracts that operate over long, defined periods (as opposed to short-lived one-off requests). The system provides information on the fulfilment state of contracts. The ability to evaluate at runtime to what extent the behaviours of the contract are fulfilled can be calculated from partial metrics. Aggregated degree of contract fulfilment can be determined from performance, cost, and service-related metrics. Fulfilment metrics can either be evaluated automatically or entered manually by a human expert. Aggregated degree of contract fulfilment can be calculated from partial metrics.</td>
</tr>
</tbody>
</table>
During execution to ensure appropriate behaviour. A mechanism can check whether the contracts to be established are achievable, given the particular properties, such as being fulfilled. From the framework specification of a given application, other requirements of the previous section can be implemented in different ways. The CONTRACT project has proposed a mapping of applications to contract technology and provided an infrastructure for the administration of contract-related processes.

The models and procedures comprising the CONTRACT framework and architecture, starting from basic concepts, are outside the scope of this paper. Where a requirement is addressed by the description we provide full technical details, including design patterns, service interfaces and agent reasoning behaviour.

In the following, we briefly describe the key design decisions of the framework and architecture and provide an overall structure of the CONTRACT architecture and framework.

**Deployment**

There is a mechanism for resolving disputes between contract parties over the third party. For security and sensitive information protection, distributed deployment should be considered whereby contracts are stored and managed in a single system and are accessed remotely by all partners. Centralised deployment is supported whereby contracts are stored with a single system and are accessed by partners. Where the target deployment configuration goals are in conflict, a hybrid deployment is recommended, where some requirements are fulfilled by centralised deployment and others are fulfilled by distributed deployment. The use cases described anticipate different ways in which the third party should be considered.
A contract store is a generalised document store with version control and contract proposals. It enforces access control on contracts so that they can only be retrieved by parties to the contract, either a specific contract ID, an accession ID, for that version of the contract document, or by searching for contracts meeting given criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to fulfil particular criteria, e.g. It may contain obligations to ful
In multi-agent systems, there has been much previous work on contract-based systems, and our approach is intended to build on, and show how our contract architecture satisfies them. In existing work, obligations are often defined at a very abstract level, leaving it to the contract parties to infer the conditions of the contract themselves from the agreements and other information.

A detailed comparison of approaches is outside the scope of this paper. However, there has been no analysis across a range of competing techniques and architectures, as well as providing a concrete context for the development of work.

In this paper, we have described the four main use cases in which our work is situated, detailing the requirements aggregated from academic projects, CONTRACT is predicated on the industrial context, and the business case from four distinct business-oriented architectures inform our work. Concepts such as norms feeding into a longer-term assessment of agents [5]. In addition, observation of fulfilment and violation of obligations potentially of contracts can integrate with other useful behaviour, such as enforcement of norms itself being a norm, are already established concrete representations of dynamic norms, management or specification (WS-Agreement).

Institutions and Norms in agent systems (including ebXML, WSLA [8], WS-Agreement [9] as well as that planned in OASIS), efforts and results to date either do not focus specifically on system specification or do not provide any formal tool support. Microsoft INDIGO platform are a case in point, since they focus on low level specification of method execution properties rather than interoperation and be compatible with, other ideas presented elsewhere. For example, the IBM WebSphere Commerce platform [10] is a Java-based high performance, multipurpose platform providing a framework for building commerce sites adequate for many different kinds of businesses, including retail, commerce and industry.

Although the WebSphere Commerce platform provides a framework for enterprise commerce, it focuses on operations and does not provide any formal tool support for the development of business rules. The software has been developed specifically for the use of developers and business analysts, and is not intended for use by legal or compliance professionals.

On the other hand, during the past decade, commercial rule engines have been increasingly used in business solutions, both for business rules and for compliance purposes. However, commercial rule engines have been developed with a focus on operations and do not provide any formal tool support for the development of business rules. The software has been developed specifically for the use of developers and business analysts, and is not intended for use by legal or compliance professionals.

In addition, the WebSphere Commerce platform provides a framework for enterprise commerce, it focuses on operations and does not provide any formal tool support for the development of business rules. The software has been developed specifically for the use of developers and business analysts, and is not intended for use by legal or compliance professionals.

The CONTRACT project seeks to develop frameworks, components and tools that make it possible to model, build, verify and reason about contracts in a systematic, repeatable and reliable way. The project is based on the idea that contracts can be decomposed into individual components and their relationships, and that these can be formalized in a way that is amenable to automated verification. The project is also based on the idea that contracts can be expressed in natural language, and that this can be formalized in a way that is amenable to automated reasoning.

The CONTRACT project is focused on the development of tools and techniques for the development and verification of contracts. The tools and techniques are intended to be used by developers and business analysts, and are not intended for use by legal or compliance professionals. The tools and techniques are based on the idea that contracts can be decomposed into individual components and their relationships, and that these can be formalized in a way that is amenable to automated verification. The tools and techniques are also based on the idea that contracts can be expressed in natural language, and that this can be formalized in a way that is amenable to automated reasoning.