# Cloud Service Localisation

Claus Pahl

Center for Next Generation Localisation CNGL and CloudCORE Research Centre
School of Computing, Dublin City University
Dublin, Ireland
cpahl@computing.dcu.ie

**Abstract.** The essence of cloud computing is the provision of software and hardware services to a range of users in different locations. The aim of cloud service localisation is to facilitate the internationalisation and localisation of cloud services by allowing their adaption to different locales. We address the lingual localisation by providing service-level language translation techniques to adopt services to different languages and regulatory localisation by providing standards-based mappings to achieve regulatory compliance with regionally varying laws, standards and regulations. The aim is to support and enforce the explicit modelling of aspects particularly relevant to localisation and runtime support consisting of tools and middleware services to automating the deployment based on models of locales, driven by the two localisation dimensions. We focus here on an ontology-based conceptual information model that integrates locale specification in a coherent way.
**Key words**: Cloud services; Service localisation; Service internationalisation; Language and governance localisation.

## 1 Introduction

Web-enabled software services, particularly in the cloud computing context, can support business and private users on a global scale [2, 6]. In regions like Europe, where a multitude of languages are spoken, services are often only developed and deployed to support a single language or region [3]. In particular smaller organisations do often not have the capacity and capability to carry out multilingual and multi-regional development. Localisation is the process of adapting digital resources like services and associated data and content to a locale, i.e. the lingual and regulatory environment (restrictions, rules, settings) of a location or region. The emergence of cloud services and the increasing trend towards end-to-end personalisation of service offerings substantiates the need to address this wider understanding of locale and localisation in a dynamic cloud service context.

The wider objectives of cloud service localisation are, firstly, to introduce cloud-centric service localisation techniques, which focus on the localisation of software and interaction at the service interface level, and, secondly, to make the localisation techniques available at runtime for dynamic cloud service localisation and end-to-end personalisation. As there are very few comparable activities, we focus in this paper on defining a conceptual information model as the backbone of a wider localisation solution. Based on this, we outline further challenges and possible architectural solutions for this context. A significant part here is devoted to the motivation of cloud service localisation as a new field of investigation.

Often, localisation refers to either languages only or physical locations only. Two different locale dimension are the focus of this investigation that embrace these and widen the concepts of localisation and locale:

- lingual localisation by providing service-level language translation techniques to adopt services (including API, description, models) to different languages,
- regulatory localisation by providing standards-based mappings to achieve regulatory compliance with regionally varying laws and regulations (business rules, standardised name/value mappings, currencies and units, and legal governance/compliance rules in relation to different regions or locations).

Progress beyond the state-of-the-art with respect to the following is aimed at:

- localisation at service interface level - classical forms of software localisation will be expanded to address internationalisation at the interface (API) level. Model mapping and translation form core ingredients to develop a coherent and integrated solution across the locale dimensions here. The challenge is a semantic model integrating heterogeneous translation, mapping and adaptation needs in a single, dynamically processable form.
- adaptation and integration - software adaptation at service-level will be considerably improved by addressing linguistic and regulatory dimensions. This poses a significant challenge, as the current focus of adaptation is only on functional and software quality aspects that are software-technical in nature.
- semantics-enhanced brokering and mediation - matching of services is equally expanded to encompass linguistic and regulatory aspects, here proposed to be included in a locale-centric negotiation process and infrastructure, requiring respective coordination protocols.

We motivate cloud service localisation in the next section. Section 3 defines the conceptual information model and its supporting architecture. In Section 4, we introduce specific localisation techniques. Section 5 discusses some challenges, before we end with a wider discussion of related work and some conclusions.

## 2 Motivation

Our focus is a platform for service localisation that makes a step from one-size-fits-all services towards end-to-end personalised service offerings based on different locales. Current cloud computing for international settings suffers from localisation and adaptability problems for multiple but different users [18, 27], which can be overcome through multi-lingual and multi-regional localisation.

### 2.1 Application Scenarios and Benefits

Various possible application scenarios can illustrate the benefits of a service localisation solution for the cloud:

– End-user media store&play cloud services. Some cloud media players are currently available in only one language. This form of end-user oriented service interaction would benefit from multilingual access by a mass market. It would involve the localisation (translation) of service description values, interaction text, and other auxiliary text. This can take place statically (prepared localisation) or dynamically (ad-hoc localisation).
– Business-targeted cloud services. A data analysis and storage service as business offerings could be adapted (localised) to be compliant with different standards and regulations for different regional or legal environments - a business-centric service offering requiring a high degree of customisation.
– End-user composed service processes. For instance, public sector applications where governance and regulations are as important as lingual aspects can be composed/configured based on several individual cloud services, adapted by individual users to their specific needs, even dynamically.
– Mediated e-commerce offerings. Telecoms-supported consumer access services are examples, where online shops avail of cloud-based support infrastructure services to allow them to manage their business online.

The second scenario shall be detailed further. A cloud-based data archiving service might need business-oriented service localisation. Sales data from local subsidiaries have to be stored centrally (e.g. in a private cloud) or usage of a low-cost storage provider abroad (in a public cloud) is envisaged. With a focus on regulatory (REG) and lingual (LING) localisation, the following is required:

– LING: translate service data between languages, e.g. from English into German - "Quote" to "Angebot" - based on standards like EANCOM or document-related attributes based on the GS1 standard for documents[1],
– REG: transform data between standards or their variants, e.g. "Quote" translates to "FullQuotation" based on a transformation between different EDIFACT variants and subsets such as EANCOM, EDIKEY, or EDIFICE. Other examples are transformations of currencies or transformation of rules and procedures, e.g. access rights to enable regulatory compliance by enabling legally required recording of activities through service adaptation.

Related data validation and data archiving services might be composed by a broker, which mediates the localisation based on integrated semantically enhanced locale policies. The requestor provides an abstract process that the broker implements - for example to offer locale-specific services abroad for roaming network customers (if compatible) using predefined mappings that are deployed dynamically. End-user composition can be applied where the end-user configures a process of different services by different services in different languages. The business searches for best provider internationally for both services, not necessarily as a

[1] For illustration, we use the EDIFACT (United Nations rules for Electronic Data Interchange for Administration, Commerce and Transport, http://www.unece.org/trade/untdid/welcome.html) and GS1 standards (supply and demand chains globally and across multiple sectors, http://www.gs1.org/).

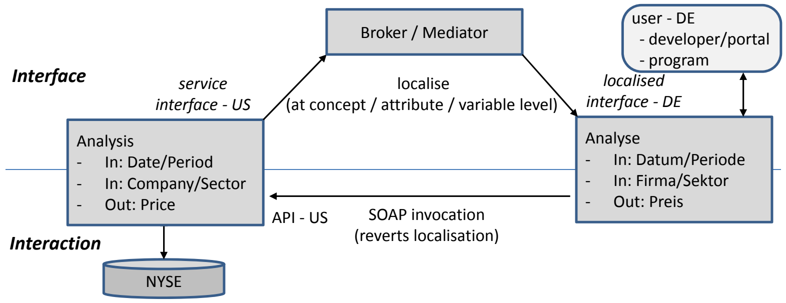


**Fig. 1.** Localisation of Stock Market Analysis Feature - Focus on Service API.

package. An example is sales records where data is validated for sector categories, e.g. GS1 for sector categorisations, across different languages.

The following illustrates a range of different localisable service artefacts with sample specifications for locales US and DE:

– API: *validate(dataset):result* in US maps to *Validiere(Datensatz):Resultat*
– Semantic Description: *compliant(GS1,dataset)* → *validated* is US maps to *einhaltend(GS1,Datensatz)* → *validiert* in DE
– Contract: *per_activation = 1 USD (incl. 18% VAT)* and *availability = 98%* in US maps to *pro_Aufruf = 1.35 EUR (incl. 21% MwSt)* and *Verfügbarkeit = 98%* in DE
– Documentation: *service validates dataset for standards compliance* in US maps to *Servce validiert Datensätze aud Einhaltung von Standards* in DE

Note that inter-artefact relationships exist: validate/validated/validates and complaint/compliance as two sets of terms that are internally cross-referenced.

Some progress has been made in the past in the context of Web service internationalisation [24, 29]. This has provided a broader framework, particularly addressing data formatting and conversion (e.g. between units) aspects. More complex, rule-based translations are, however, not included.

### 2.2 Use Case and Requirements

In addition to the demonstration of the need for localisation across a wide spectrum, we chose another use case to elicit more detailed requirements. We use an environment that provides service-level access to stock market information and analyses[2]. A German user might want to access data from the New York stock exchange, which is provided in an English format. We present a scenario in which the user can implement a locale-compliant interface, i.e. one that allows technical interaction of service interface and description aspects in German.

At the application-level, two sample calls of a stock market data analysis service for the two locales (US-locale with English as the language and USD as

[2] This is based on a case study using financial stock market information services from http://xignite.com/ and http://deutsche-boerse.com.

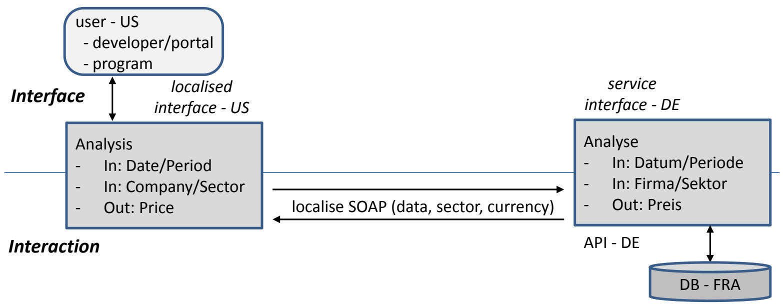


**Fig. 2.** Localisation of Stock Market Analysis Feature - extends NYSE-specific Analysis Service to include DB-FRA.

currency and DE-locale with German as the language and EUR as the currency) could be: *Analyse(10/30/2011, logistics)* $\rightarrow$ *3.82 USD* and *Analysis(30.10.2011, Logistik)* $\rightarrow$ *4.23 EUR*. Localisable artefacts in this example are

– Date: a format change is needed - which would also apply to time and collation issues,
– Sector: data values describing an industry sector are localised based on a translation between standardised terminologies - which would also apply to product categories,
– Language: operation names (and possibly other interface and model elements) are translated between languages,
– Currency: values are converted - as would be other measurements and units.

This list can be extended: different regulatory environments based on maybe multilingual and standardised glossaries and dictionaries; calculations/conversions based on rules (fixed) or repositories (dynamic); tax rates and customs duties can be added if products are sold; any messages including help and error messages to which text translation would be applied. Typical examples for technical terms that need translation in the banking or stock markets context are (average price - Durchschnittspreis), (main trading phase - Haupthandelsphase), or (volume weighted average - volumengewichteter Durchschnitt) that are based on accepted, often standardised terminologies. Some examples might be defined in terms of classification and categorisation standards: (logistics - Logistik) for a sector or (dairy - Milchprodukte) for product categories.

In the example, the user-DE can discover services based on a German specification and can invoke them based on a German interface. A stock market analysis provider can add a DE-locale to its default US-locale policy. This would result in a correct match in a full negotiation process in which a user searches for services that are provided in a locale-specific way since the provider is able to support US-to-DE locale mappings if required. In an architecture that implements these localisation translations, service instrumentation would result in a process to be generated and enacted, rather than a single SOAP request as indicated in Fig. 1. This process could comprise service invocation and logging (location) for accountability where the location is a parameter, which indicates where and how records are kept (if ruled by privacy laws). A coordination protocol then governs the exchange of locales and the following SOAP-based application interaction.

The above scenario could be further extended to allow an American user (with locale US) to access a German-language stock market information provider, e.g. for Deutsche Börse, Frankfurt. Localisation can be provided on a SOAP level primarily, see Fig. 2, for automated service activations and interactions.

## 3 Conceptual Framework

The localisation of possibly brokered and mediated cloud services is the application context in which service localisation takes place - requiring a coherent integrated information model for multi-dimensional localisation and supporting cloud infrastructure to deal with localisation statically and dynamically in an automated form. Our aim is the localisation of the (automated) interaction with software services as dynamic, executable artefacts. What is required to make both software-to-software and user-to-software interaction localisable is interface-level dynamic localisation through translation and adaptation techniques. The conceptual framework we present is comprised of an information architecture, which is essentially an ontology, a systems architecture and an abstract process that would be enacted on the architecture [19, 20].

### 3.1 Information Architecture - Localisable Artefacts

Localisable artefacts are service specifications and models, but also contracts and documentation addressing aspects like interaction in the form of messages and also licensing, usage and access rights, see Fig. 3. Translation between controlled vocabularies is required. If services are not locale-independent, localisation is needed. A locale model is based on locale-affected attributes like time/data format (the localisable artefact is data), language (the localisable artefact is interaction - text and/or dialogues), and collation/ordering (the localisable artefact is data). The user-specific locale models (user locale policies) are applied at the service endpoints after a locale negotiation process with the provider. Services are described in terms of their interfaces (in WSDL): data types, data, operations; more comprehensive models (e.g. the Unified Service Description Language USDL - http://www.w3.org/2005/Incubator/usdl/): functionality, quality, locale (language, currency, units, legal/tax), context (location, device, platform), and documentation (manual, help). Consequently, localisable artefacts are

- service interface: data-level internationalisation and transformation for the different locale dimensions that applies in particular to data (schema and data-level translation based on EAN Identification, EANCOM/EDIFACT, etc.) and mappings; RESTful HTTP for instance natively supports content-type negotiation for interactions (messages).
- semantic service description and contract: aspects such as licensing, usage and access rights, but also other metadata aspects are subject to translation between controlled vocabularies (data-level translation).

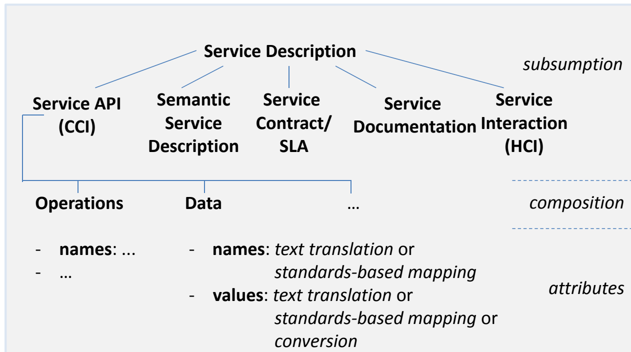


**Fig. 3.** Localisation Artefact Ontology - Excerpt with Focus on Service API.

These service descriptions are linked to the generic service description concept through a subsumption relationship. Each of the description types is then decomposed (using an is_part_of relation) into individual description elements, which are subject to localisation. Composition is necessary as a relationship as the subsumption (subtype) idea does not apply between an artefact and its parts. For the service API, operations and data (in the form of input and output parameters or messages) are these individual, localisable elements. Each localisable element has attributes that characterise the localisation technique applied to it. We have indicated their localisation type, which can be text (to be translated using usual MT techniques), standards-based data (where the translation is defined in a predefined glossary) or conversions (e.g. for statically defined measurements or dynamically defined currency conversions). For the given ontology excerpt, the translation type attribute is given. Composition and attributes are intra-artefact properties. There are also inter-artefact properties: is_defined_in and is_explained_in are examples for cross-artefact properties, e.g. a contract refers to a definition of an operation in an API or data is explained in documentation.

The localisation ontology is the foundation, on which later on rules to define localisation policies and actions to be executed for localisation will be introduced.

### 3.2 Systems Architecture and Process

A service localisation solution consists of guided translations (through ontologies), which may be pre-translated. Static value mappings (cross-language or cross-regulation) and dynamic value mappings need to be combined. A static setting means that localisations are prepared. A system architecture should allow developers to prepare material for multiple locales in advance and check their quality. An intermediary then deploys and executes respective techniques depending on user profile and negotiation, from which SLAs are formed. These are used to govern the invocation of services. In a dynamic setting, an intermediary selects suitable services (involving query translation), carries out negotiation based on best mappings (closest profiles) by using localisation and quality assur-

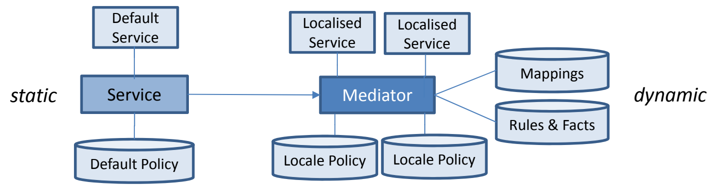


**Fig. 4.** Localisation Systems Architecture.

ance services. Overall, the achievable quality will vary between in-advance and fully-automated scenarios. Corresponding assurance levels will need to be set.

These techniques can be facilitated for service localisation through process adaptation and instrumentation [26]. An intermediary mediates between several clients (at different locations) and several providers (at different locations) by providing this core process with localisation adaptations [23]. Several localisation patterns emerge. In a single-provider setting, one provider supports n (n≥1) locales; in a market-place setting, one client uses a composition of provider services (n>1). Both enact a localisation process based on locale negotiation (brokering) and localisation (mediation) implementing a localisation coordination protocol.

– Dynamic Generation: for regulatory localisation, aspect-oriented instrumentation can be generated on-demand per defined locale, allowing the user control over locale definition
– Configuration Management / Generation: different endpoints (and respective bindings) for different locales are generated for a localised API to which then localisations (translations and mappings) are applied; which of these to use might be dynamically decided
– Negotiation and Coordination: exchange and agreement on locale policies through SOAP headers based on [26] for coordination
– Architecture: data/information integration layer based on ontologies; service localisation layer based on adaptation (user model) and regulatory and lingual localisation; and management through locale negotiation

The process that governs the overall activities is *Negotiation*, followed by *Mediation & Localisation* and *Execution*. Abstract services and locale policies are the negotiation basis, based on which service endpoints give access to application services or instrumented processes, which in turn needs a basis of core services for translation LING, adaptation REG, and necessary instrumentation resulting from these, i.e. is localisation as a service, composed into a process.

## 4 Localisation Techniques for the Cloud

### 4.1 Policy Language and Localisation Model

At the core of the localisation techniques is a rule language [16, 25, 28]: an interoperable locale policy language is needed in addition to languages like WSDL, which captures the functionality side of the service API, to define locale descriptions for both provider and user. The policy language is based on Semantic

Web rule principles with underlying ontology support (OWL) for the conceptual aspects. A locale policy defines the individual rules and instrumentations that characterise a locale. In Fig. 5, we define a layered localisation model that connects locales and rule-based locale mappings:

1. An ontology-based base layer captures the different translation types. The ontology defines the localisable artefacts and their relationships (see Fig. 3). Each unit is characterised in terms of a number of attributes:
   - translation unit type: API (data, operation), semantic description, SLAs, or documentation are examples.
   - translation type: standards-based, ad-hoc text translation, conversion based on distinct repositories, called translation memories (TMs) here.
2. Basic rules are provided, like *Locale* or *hasCurr*, on which specific locales can be defined. An example is

   $$UKLocale(?l) \leftarrow Locale(?l) \wedge hasCurr(?l, ?c) \wedge ?c = GPB$$

   Higher level rules allow a locale to be inferred from partial knowledge, e.g. $?c = GBP \rightarrow ?l = UKLocale$ and to detect inconsistencies.
3. Depending on provider and consumer locales, translations might need to be executed. Translations are guided by the locale definitions and use mappings defined in the translations layer (either as pairwise translation units (e.g. text) or as executable conversions (e.g. for measurements or currencies). The overall translation is dynamically assembled from the translation base (memory). The conversion rule for currencies can be dynamically created:

   $$UKLocale2DELocale(?l1, ?l2) \leftarrow hasCurr(?l1, ?c1) \wedge hasCurr(?l2, ?c2) \wedge \\ ?c2 = convertCurr(GBP, EUR, ?c1)$$

   Mappings appear in a primary form as a translation between locales. However, in some cases, consequential actions need to be added. For instance, dynamic currency conversions need to be added or logging of activities for compliance reasons needs to be added. Actions thus comprise another type of activity than translation. Localisation might entail additional activities such as adding logs for activities.

Locale configuration examples are the metric system of units or a tax system. A locale mapping is compliant if it does not violate any locale configuration. While a coherent conceptual model has been presented in Fig. 5, specific support is needed for the two main localisation dimensions:

- Linguistic localisation and machine translation: the challenge is the localisation of service artefacts with little textual context - at design and run-time.
- Regulatory localisation and governance: integrated and coherent adaptation to different regulatory standards and procedures based on semantic, rule-based techniques using multi-locale modelling and discovery using ontologies; and multi-dimensional user and service models and mappings for adaptation.

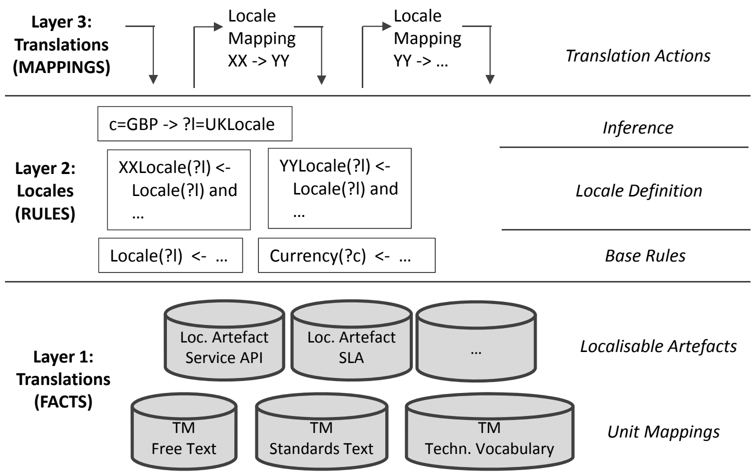


**Fig. 5.** Conceptual Localisation Model.

Consistency is a key motivation behind the rule-based approach. Locale mappings need to be consistent. Inference rules in the rule layer provide the mechanism to detect inconsistencies and enforce correct mappings. An example is that countries use a coherent set of measurements, e.g. the metric system. Also, currencies are linked to fiscal systems defined in a specific language. Since locale mappings are often dynamically assembled from individual translations, their composition needs to be checked for consistency.

### 4.2 Lingual Localisation and Translation

Localisation of artefacts with little textual context - in prepared and dynamic settings - is the challenge. We propose the translation of technical content based on reduced-context machine translation techniques [12]. We argue that specific techniques are needed to enable reliable translations between formalised, technical content (such as ontologies, service API and service models) [11]. Multi-lingual ontologies form the core of the technique by providing a mapping / glossary (as core translation memory). To enable effective processing, there is a need to consider the merging of prepared translation material with other translations generated on-the-fly. The key concern is the accuracy and quality of the translation, which needs to be trustworthy in automated processing environments.

The specific solutions are ontology translations of technical and business application domains and translation of controlled technical context with little contextual information [12, 13]. The outcome consists of translation techniques and supporting translation memories using predefined units (words/phrases) for standardised multilingual glossaries and variable units (words/phrases) - using statistical machine translation (SMT) techniques based on ontological proximity to guide the free translation. XLIFF (XML Localization Interchange File Format), an OASIS standard for localisation exchange - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xliff), is used to formulate translations. Translations units (pairs) are kept in translation memories. A further investigation is beyond the scope here.

### 4.3 Regulatory Localisation and Governance

Regulatory localisation and governance through adaptation to different regulatory standards and procedures is based on localising regulatory concerns, which are often captured in terms and conditions. Regulations apply for instance with respect the identification and description of business entities or the way processes are handled [17, 26]. Furthermore, regulatory locale aspects like tax, currencies and units need to be addressed [4, 24].

The specific objective here are multi-facetted rules that are modular and composable and that enable interference checking. For data object localisation, we argue here that a multilingual, multiregional schema mapping and integration technique is needed to adapt information to regulations in different regions (in possibly different languages). For SLA translation, we need to enable mappings of a SLA into a different locale that needs to consider the respective standards and procedures. A translation technique needs to ensure quality assurance and accountability of translation. We distinguish physical units like length, size, weight, but also colour and financial aspects like pricing and payment, currency, or tax. The outcome is a set of regulatory rule-based information translation techniques with predefined mappings using standards conversions between lengths and other units and consequential actions (instrumentations) needed when dynamic conversions (e.g. currency) or auxiliary actions (logging) are entailed. Individual definitions are not as much the problems as is keeping consistency.

## 5 Infrastructures for Cloud Localisation - Directions and Requirements

Two central infrastructure and mediation solutions are needed:

- Coordination: a localisation coordination strategy and protocol to implement the locale negotiation and coordination of localisation for different locales - based on a rule-based solution with a conflict resolution strategy.
- Service Mediation: a brokering and mediation architecture where an intermediary mediates between clients and providers at different locations. Solution components are based on hybrid techniques for guided translation, model mappings (cross-language/regulation) and process adaptation techniques.

We discuss possible directions and techniques here.

The first aspect is semantic technologies for service localisation. Multi-locale modelling and service adaptation using ontologies needs to provide a new approach to semantic service description through locale-specific domain ontologies. Locale-specific domain models are needed. Ontologies provide the formal framework. To define such ontologies, there is a need to consider a data layer and a services layer, each with respective semantic annotations supported. A *multi-lingual, multi-ontology framework* with corresponding facts, rules and mappings is needed [9]. We suggest to adopt the *linked data* approach followed by other service modelling approaches (like USDL - http://www.linked-usdl.org/). The XLIFF translation mapping standard can be converted into RDF.

The second aspect is software brokering - adaptation and mediation of services. Multi-dimensional user and service models and mappings need specific techniques for service adaption and personalisation. A user model developed around a notion of locale is needed to achieve cross-lingual and cross-regional interoperability. To match user models and service profiles, there is a need to formalise and automate mappings between the different models. On the regulatory localisation side, trustworthy conversions between formats and regulations need to be defined, based on guidance from ontologies. On the lingual localisation side, format/meta data translations and translation of values and text need to be facilitated, in both guided and unguided forms.

Brokering and mediation architectures provide a framework for locale mediation based on negotiation and localisation implementation activities [21, 22]. We argue here that a *dynamic localisation engine* is needed. This is made up of services providing discovery, locale negotiation (static and dynamic, to generate SLAs and BPEL code, respectively), localisation (lingual and regulatory, prepared and on-the-fly, with quality check). To support this engine, there is a need to define a *localisation process* to govern the individual activities [15]. A repository is proposed to ensure the efficient and effective operation of the engine, which maintains pre-translated content and data/format mappings as translation memories. The outcome is a set of services to negotiate and localise and a repository to keep prepared localisations and schema information.

A locale is a specific type of a context. *Context- or locale-driven service provision* is the aim. From a cloud provider's perspective, this can also be seen as a *multitenancy* problem, where each cloud tenant is defined by its locale. Both aspects are further discussed in Section 6.

## 6 Related Work

Several areas are related to the subject here, although we provide a different perspective compared to other research publications [10, 16, **?**,28] and work covered by related EU-supported research projects, like SOA4All (www.soa4all.eu), ACSI (www.acsi-project.eu/), 4Caast (4caast.morfeo-project.org/) or mOSAIC (http://www.mosaic-cloud.eu/), which address end-user adaptivity using generic semantic models (SOA4All), software-centric coordination and marketplaces (ACSI, 4Caast), or multi-cloud provisioning (mOSAIC). Our framework is orthogonal to these efforts towards end-to-end offerings, and unique in its interdisciplinary character focusing on linguistic and regulatory aspects. More general, our proposal relates to the following aspects:

- Software Localisation: localisation of software normally refers to the human consumption, i.e. messages and dialogues produced by the software. We focus on localisation at the service interface level through internationalisation and ontology mapping and translation. The current technology on service internationalisation, which is the closest, is supported by the W3C Service Internationalisation activity [24, 29]. The focus there is data level localisation, specifically for dates, currencies and units and common approaches to collation. The necessary solutions are conversions - e.g. statically defined between units or dynamically defined between currencies. Lingual aspects or more complex regulatory or business aspects are not covered, although for instance taxation as a sample concern is mentioned.

– Adaptation and Integration: software adaptation at service-level and data integration are common techniques. An example are schema mappings for data integration, where consistency and semantic preservation are key concerns. We follow ontology-based approaches, using these for semantics preservation, but add a multi-lingual layer using ontology mappings. In a similar context, service/process instrumentation is used to add enhanced processing abilities, but as for data integration and adaption, our focus on lingual and regulatory concerns within a rule-based framework is new.
  Context-awareness of services is a direction that has been covered by a range of contributions [4]. The notion of context is similar to that of locales, reflecting properties of the execution or client environment. However, context usually does not include lingual or regulatory aspects. We propose to adopt and extend respective adaptation and instrumentation techniques [5, 14].
  Multi-tenancy is a cloud computing problem [18, 27] that requires solutions for different users with different needs to be kept separate.
– Semantic Technologies: matching of services and supporting the negotiation process and infrastructure [7, 10, 19]. Through ontology mappings, multi-lingual terminologies and multi-regional regulations can be captured and dynamically processed.

## 7 Conclusions

Service localisation for cloud computing is a form of service personalisation and adaptation. This focus addresses service engineering principles, methods and tools by allowing service to be adapted to different locales. Services are enabled for seamless integration by providing localisation of services as software in conjunction with related content/data processed and communicated by these services. We have discussed techniques to enable the expansion of service offerings into different markets for cloud solution and application providers, which is a context of significant economic advantage [1, 8]. Localisation is a means to bring products and services to markets that are otherwise inaccessible.

Multi-locale services support interoperable clouds contributing to a market of services by allowing services to be internationalised and localised - an aspect of crucial significance for the EU with 27 members, and even higher numbers of local languages and regulatory systems. Service localisation adds to the availability of platforms for easy and controlled development and deployment of value-added services through innovative service front-ends by providing cloud infrastructure services to localise and manage multi-locale services. Localisable services enable lower barriers for service providers and users to develop, select, combine and use value-added services and to allow providers to enter new markets, particularly SMEs without in-house localisation capabilities.

The objectives of this paper were two-fold. Firstly, we presented a conceptual framework to capture the key concerns of multi-lingual, multi-regulatory service localisation as a rule- and ontology-based information model. Secondly, our discussion here aimed to motivate the need for research in service localisation for the cloud to be carried out and analyse the major concerns. In this vein, we tried to identify some directions and concerns to be addressed. The focus of our investigation was, inevitably, limited: program-level localisation, end-to-end cloud personalisation and truly multi-lingual clouds are examples of omissions - which we intend to investigate further. It is also clear that implementation work needs to be done and evaluated - so far, the conceptual solution is only motivated and justified through the discussed case studies.

## Acknowledgment

This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation at Dublin City University.

## References

1. 451 Group. Report on Cloud Computing 'As-a-service' market sizing - Report II (2010)
2. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. Comm. ACM 53, 4. pp.50-58. (2009)
3. EU Commission. Report on "The Future of Cloud Computing" - Opportunities for European Cloud Computing Beyond 2010. EU. (2010)
4. Bandara, K. Y.,Wang, M. X., Pahl, C. Context modeling and constraints binding in web service business processes. In Proceedings of the Workshop on Context-Aware Software Technology and Applications CASTA. ACM Press, pp. 29-32. (2009)
5. Baresi, L., Guinea, S. Towards Dynamic Monitoring of WS-BPEL Processes. Lecture Notes in Computer Science, vol. 3826. Springer, pp. 269-282. (2009)
6. Buyya, R., Broberg, J., Goscinski, A. Cloud Computing - Principles and Paradigms. Wiley. (2011)
7. Doulkeridis, C., Loutas, N., Vazirgiannis, M. A system architecture for context aware service discovery. In: ENTCS, pp.101-116. (2006)
8. Fingar, P. Cloud computing and the promise of on-demand business innovation. Intelligent Enterprise. (2009)
9. Fu B., Brennan R., O'Sullivan D. Multilingual Ontology Mapping: Challenges and a Proposed Framework. In: Workshop on Matching and Meaning - Automated Development, Evolution and Interpretation of Ontologies. (2009)
10. Fujii, K., Suda, T. Semantics-based context-aware dynamic service composition. ACM Trans. Auton. Adapt. Syst. 4, 2, pp.1-31. (2009)

11. van Genabith, J. Metaphors, Logic and Type Theory. Metaphor And Symbol, Vol. 16, No. 1/2, 2001, Lawrence Erlbaum Publishers, ISSN 1092-6488, pp.43-57. (2001)
12. van Genabith, J., Crouch, R. 1999. Dynamic and Underspecified Semantics for LFG. In: Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach; (ed.) Mary Dalrymple; MIT Press, pp.209-260. (1999)
13. Graham, Y., van Genabith, J. An Open Source Rule Induction Tool for Transfer-Based SMT. In: The Prague Bulletin of Mathematical Linguistics, Special Issue: Open Source Tools for Machine Translation, 91, pp.37-46. (2009)
14. Kapitsaki, G., Kateros, D., Prezerakos, G., Venierris, I. Model-driven development of composite context-aware web applications. Information and Software Technology 51, pp.1244-1260. (2009)
15. Karastoyanova, D., Leymann, F. BPEL'n'Aspects: Adapting service orchestration logic. In: International Conference on Web Services. (2009)
16. Kharbili, M.E., Keil, T. Bringing agility to business process management: Rules deployment in an SOA. In: 6th IEEE European Conference on Web Services. (2008)
17. Leusse, P.D., Dimitrakos,T., Brossard, D. A governance model for SOA. In: IEEE International Conference on Web Services. (2009)
18. Mietzner, R., Unger, T., Titze, R., Leymann, F. Combining different multi-tenancy patterns in service-oriented applications. In: IEEE International Enterprise Distributed Object Computing Conference (2009)
19. Pahl, C., Zhu, Y. A Semantical Framework for the Orchestration and Choreography of Web Services. International Workshop on Web Languages and Formal Methods WLFM'05. ENTCS (2005)
20. Pahl, C. A Formal Composition and Interaction Model for a Web Component Platform. In: Workshop on Formal Methods and Component Interaction FMCI02 (2002)
21. Pahl, C., Giesecke, S., Hasselbring, W. An Ontology-based Approach for Modelling Architectural Styles. In: European Conference on Software Architecture ECSA2007. Springer-Verlag, LNCS Series (2007)
22. R. Barrett, L. M. Patcas, J. Murphy, C. Pahl. Model Driven Distribution Pattern Design for Dynamic Web Service Compositions. In: International Conference on Web Engineering ICWE06. ACM Press (2006)
23. Pahl, C. Dynamic Adaptive Service Architecture - Towards Coordinated Service Composition. In:European Conference on Software Architecture ECSA'2010. Springer LNCS (2010)
24. Phillips, A.P. Web Services and Internationalization. Whitepaper. http://www.inter-locale.com/whitepaper/multilingual/ml73-ws-20050524.xml (2005)
25. Rosenberg, F., Dustdar, S. Business rules integration in BPEL - a service-oriented approach. In: 7th IEEE Intl Conference on E-Commerce Technology (2005)
26. Wang, M.X., Bandara, K.Y., Pahl, C. Integrated constraint violation handling for dynamic service composition. In: IEEE International Conference on Services Computing (2009)
27. Wang, M.X., Bandara, K.Y., Pahl, C. Process as a Service - Distributed Multi-tenant Policy-based Process Runtime Governance. In: IEEE International Conference on Services Computing SCC 2010. IEEE Press (2010)
28. Weigand, H., Heuvel, W.-J. v. d., Hiel, M. Rule-based service composition and service-oriented business rule management. In: Interdisciplinary Workshop Regulations Modelling and Deployment (2008)
29. W3C. Web Services Internationalization Usage Scenarios. http://www.w3.org/TR/ws-i18n-scenarios/ (2005)