# E-Government for Distributed Autonomous Administrations

**Daniela Feldkamp[1], Wolf Siberski[2], Barbara Thönssen[1], Holger Wache[1]**

[1]University of Applied Sciences Northwestern Switzerland
Riggenbachstrasse 16, 4600 Olten, Switzerland
daniela.feldkamp@fhnw.ch
barbara.thoenssen@fhnw.ch
holger.wache@fhnw.ch

[2]L3S Research Center
Institut für Verteilte Systeme, Appelstraße 4, 30167 Hannover , Germany
siberski@L3S.de

### Abstract

E-Government is no longer a concern of single public administration units. State wide E-Government strategies, E-Government architectures and frameworks are established now and public administrations are willing to suit the action to the word. Also, semantically enriched techniques for service description, discovery and invocation are ready to come into practice.

The challenge is the appropriate use of all the relevant parts to meet the technical, organizational and legal requirements.

The paper at hand provides an approach, how available techniques can be implemented in a light-weighted way to build 'one-stop E-Government services' on completely decentralized components. This allows to leave the control to where it belongs (every service contributor is responsible for its parts), to execute a service according to the local ICT but at the same time to meet the requirements for automated cross-administration cooperation.

The introduced approach is illustrated with the example of Switzerland.

## Introduction

Having one-stop E-Government services is a major goal of any E-Government strategy[1]. In the last few years huge effort has been put into clearly structured service description, service (re-)engineering or even service automation[2]. Nevertheless, many services can not be provided by one but by various organizational units within the same or different public administrations. Therefore, to perform one-stop E-Government services cross-administration processes must be implemented.

However, modeling, hosting, ownership and invocation of cross-administration processes are still open problems. As in federal states like Switzerland or Germany the different public administrations (in Switzerland municipalities, cantons and the state) are granted huge autonomy by law and no superior authority exists or can be set up. Without a superior authority, the question arises who will model, own and host the cross-administration services accepted by all organizational units. Furthermore, every single cross-administration service is a matter of negotiation and commitment, e.g. sharing IT infrastructure for the reason of cost cutting, improving service quality by focusing on unique services, or simply combining services of different units. Even though cross-administration services are a strategic goal, local administrations hesitate to change or even to adapt their business processes accordingly as this always means additional costs.

However, a first step into collaboration is to make the local services available for invocation by others. Traditional service architectures suppose a central instance establishing a single point of service registration. In Switzerland it fails because no superior authority is pushing and supporting the cooperation and maintenance. Practice shows that without a binding commitment updates are not reliable and without that no system can perform properly.[3]

[1] amongst others: IT-, eGovernment-und Multimedia-Strategie für die Landesverwaltung [Rheinland-Pfalz] 2006 – 2011. http://www.zukunft.rlp.de/ITZ/nav/4d0/binarywriterservlet?imgUid=d1970c31-09ae-a211-9599-06a90fb0e223&uBasVariant=11111111-1111-1111-1111-111111111111&isDownload=true. (10/18/2007).
eGovernment Factsheets. eGovernment in Spain. July 2007. Version 0.8. http://www.epractice.eu/files/media/media711.pdf (10/18/2007).
E-Government Strategie Schweiz. http://www.isb.admin.ch/themen/egovernment/00067/index.html?lang=de&download=NHzLpZeg7t,lnp6I0NTU042l2Z6ln1acy4Zn4Z2qZpnO2Yuq2Z6gpJCDdIJ6fGym162epYbg2c_JjKbNoKSn6A-- (10/18/2007)

[2] In Switzerland the association ‚Verein eCH' develops standards, best practises and resources for E-Government. URL: http://ech.ch/index.php?lang=en (10/04/2007). There is one special working group concerned with business process management: In 2007 a new standard for process modeling, along with best practices and guidelines, have been proposed to support public administrations in their efforts.

[3] The state portal www.ch.ch has been set up about 5 years ago to guide users to the web-sites of that public administration able to handle the user's request. Therefore, Public Administrations were asked to provide and *maintain* the respective information. Even though the portal was hosted by the state without costs for the cantons or municipalities

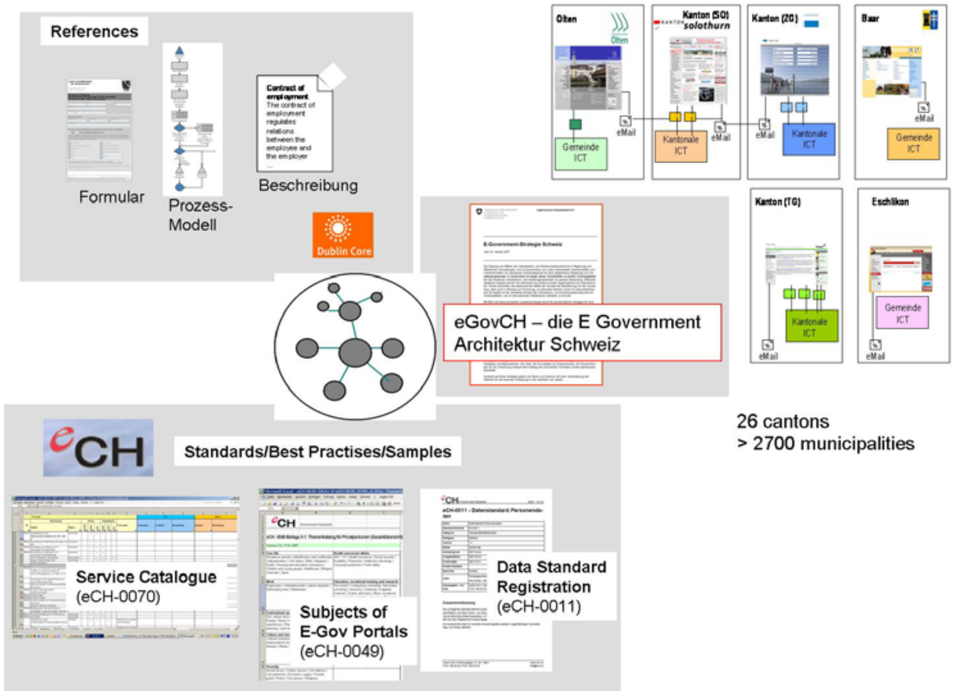


**Figure 1: E-Government components for Public Administrations**

A more promising approach is to reduce agreement to an absolute minimum and allow for the maximum of independence and autonomy. Already defined standards, e.g. for data exchange, could smooth the path.
Semantic Web Services (SWS) (e.g. (Studer et.al. 2007)) aim to simplify the modeling and enacting processes in a cross-company manner. Focussing on the semantic level they weaken the problem of negotiating. But current proposed languages, i.e. OWL-S or WSMO (see (Studer et.al. 2007)), for building semantically enriched services come with a huge modeling effort. Many business practitioners have problems to model their services and are not able to use all modeling concepts accordingly. Furthermore, current SWS approaches lack a reliable enacting environment. These problems prevented the use of SWS in a real-world application. This is especially true for the scenario of one-stop E-Government services in the introduced environment: Semantic Web Services fail because there is no central authority to take the position of a knowledge engineer capable to model, host and perform cross-administration processes. Furthermore classical SWSs fail because they still assume a central repository where all (semantically enriched) web services are described.

Therefore, in this paper we propose a more light-weighted approach, characterized by

- No need for modeling, hosting and maintaining of cross-administration processes.
- No central point of service registration. All services are registered at a local instance
- Interaction between different autonomous units based on a minimum of commitments. A very small ontology describes the subject of services with their minimal set of inputs and outputs. The ontology is shared by all units.

In Switzerland such an ontology can be derived from an already committed ontology set up in the context of term standardization. Cross-organizational services can be invoked in a fully distributed environment based on the broker concept. A broker provides the entry point to an organizational unit and starts all registered services of one organizational unit. Brokers of all participating units are connected in a scalable peer-to-peer architecture without any central control.

The paper at hand is organized as follows: First, characteristics and the current state of E-Government affairs, with respect to one-stop services, are outlined, a use case is introduced and generalized requirements are derived. Then, the suggested solution is explained comprising the broker functionality and ontology structure. In the following section, we describe how to exploit a peer-to-peer-architecture for service publication and lookup, and describe how the discovered services are invoked. The next section gives an example of service implementation in Switzerland. The paper closes with the identification of benefits of the approach.

---

maintenance has been done poorly and in the meantime a project is set up to automatically retrieve the necessary information.

## E-Government in Switzerland

States with a distinct federal structure like Germany or Switzerland face specific problems when implementing cross-administration services. As there is no superior authority to define, model and operate such services the focus is laid on common standards, shared resources and definitions of interfaces[4]. The framework is given by the E-Government strategy (in Switzerland by the Informatikstrategieorgan Bund, 2007), completed with service architecture blueprints (Müller 2005).

Currently, the Swiss Federal Chancellery is working on a project to provide so called references, e.g. standard texts for service descriptions or form templates, to all public administrations concerned. Meta data for these references are based on Dublin Core. In turn, the Dublin Core terms are part of the recently developed 'E-Gov Upper Ontology', a small ontology about the most important terms and their relations to describe the E-Government domain in Switzerland.[5]

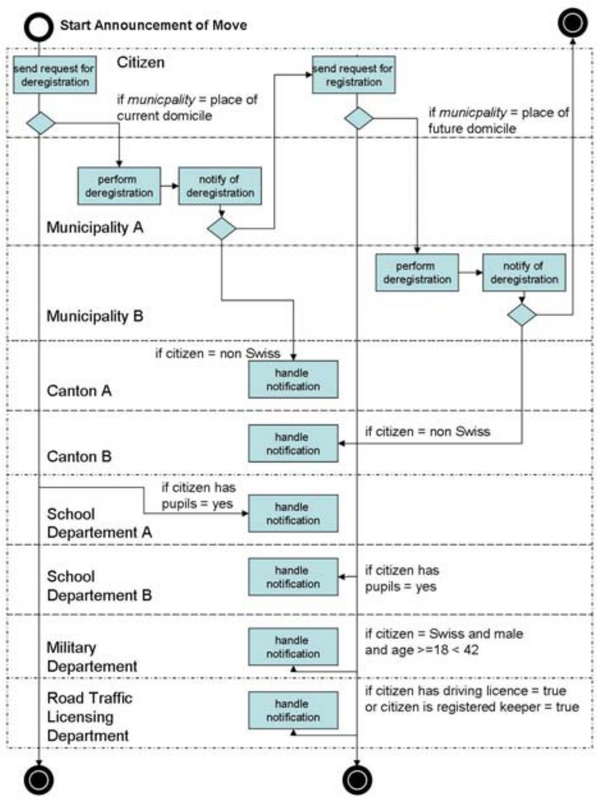


**Figure 2: Tasks to be performed within the 'virtual process' of an announcement of move**

4 In Switzerland the association ‚Verein eCH' develops standards, best practises and resources for E-Government URL: http://ech.ch/index.php?lang=en (10/04/2007)

5 The ‚E-Gov Upper Ontology' has been developed within the scope of a project authorized by the Informatikstrategieorgan Bund and accomplished by students of the University of Applied Sciences Switzerland

### The missing link

All of the above mentioned components (depicted in Figure 1) aim to resolve barriers to E-Government cross-administration service provision. However, none of it can become active to really *build* and *run* a service.

On the public administration's side more and more have (re-)engineered their internal services reaching a clear understanding of what input and output data is mandatory, what interfaces have to be served and how the process has to be performed. Though service execution can highly vary from fully automated processes to paper based handling depending on the public administration's ICT. Even though SOA is the goal, public administration's ICT is still based on large or medium sized, monolithic systems or in best cases client-server-architecture. Automated service invocation, e.g. where data is entered in an application form provided on the web and processed in a backend system is still rare[6]. Cross-administration services are mostly performed individually depending on special arrangements between the involved organizational units and throughout non-automated. Who has to be involved under which circumstances is on the one hand managed by the individual administration on the other hand ruled by state or canton-wide law and regulations. A centralized approach for cross-administration service modeling would require a huge effort of reconciliation between – in some cases – all of the municipalities, cantons and the state.

Therefore, an alternative is needed to link the internal services independently and 'on demand' to perform automated cross-administration services.

### Use case: 'Announcement of Move'

One-Stop E-Government services require the contribution of more than one public administration, e.g. different organizational units within one public administration, two or more municipalities, a municipality and a canton etc. One of these cross-administration services is 'announcement of move'. If a person in Switzerland moves he or she has to deregister in one municipality and register in another. In certain cases the canton(s) have to be notified, in others a school authority must be involved or a second registration for a car or a pet must be made. In some municipalities data about health insurance must be provided, in others not. It should be noted that in Switzerland it differs from municipality to municipality which services have to be invoked. This makes also one-stop E-Government services very difficult because

6 The most important reasons for that were investment savings. In the 80s[t] and 90s[s] core business tasks have been costly automated but increased efficiency, reliability and quality significantly. These systems still work fine as only minor business changes have taken place since. For example, one still needs the same data to handle a building application, register a child's birth etc. As a citizen can not choose another 'supplier of building permissions' there was no real pressure on Public Administration to change a running system.

Switzerland has more than 3,500 different organizational units, each performing its own set of (de)registration services.
Today, some of the tasks have to be triggered by the citizen (e.g. the citizen is responsible that deregistration in one municipality and registration in the other is performed and, if applicable, the school authority is notified or car and pet are registered), others by the municipality (like pre-information of the municipality the citizen moves to, notification of the canton in case a non Swiss person moves, notification of the tax authority).
Even though some municipalities provide forms for (de-)registration on the internet or even import this data into their registration system, cross-administration service 'announcement of move' is nowhere available.
To have an automated one-stop E-Government service, all service participants must contribute if requested. Therefore, it must be ensured that all the relevant sub-services are detected and performed (only) by the responsible public administration.
Figure 2 gives a graphical representation of the above mentioned tasks constituting the whole process of announcing a citizen's move.

## Requirements

This goal requires that all service providers contributing to cross-administration services must describe their internal services and data in a machine understandable way, considering that

- mandatory data is standardized (or at least negotiated between all contributors)[7]
- mandatory input for all participating internal services is defined
- interfaces to the internal services are defined; as the local ICT varies that much, full integration of the internal service must be possible as well as loose, e.g. invoking the internal service via e-mail notification

In addition it must be ensured that every public administration

- remains responsible for its internal services
- is free to add or remove internal services from cross-administration service provision at any time
- is considered in any case a service is requested
- is able to execute its internal processes in its way
- is able to distribute their services to other public administration for execution.

[7] In case of the 'announcement of move use case a standard can be used: 'eCH-0011 Data Standard for Registration'. URL: http://ech.ch/index.php?option=com_docman&task=cat_view&gid=146&Itemid=181&lang=en (10/04/2007)

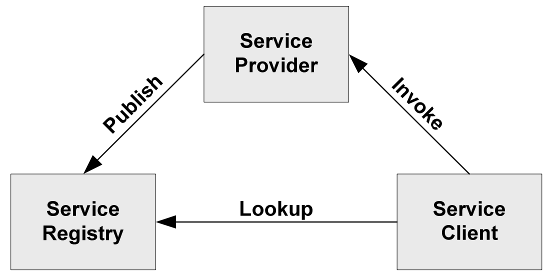


**Figure 3: Standard Web Service registration and discovery**

## E-Gov Broker

In opposition to most service architectures which base on a centralized registry we propose a decentralized approach with loose coupled components. Like the centralized approach, our decentralized proposal also has to perform three activities related to a service invocation (see Figure 3):

- *Service Publication*: The service provider publishes the service specification at a registry so that it can be found later on.
- *Service Lookup*: The service client searches for suitable services based on some constraints
- *Service Invocation*: The service client executes the service

In standard web service scenarios, the availability of a central registry is assumed (Bellwood et al. 2002). While this simplifies service lookup, it also requires central coordination of all services, which is not available in our context. We need an approach, which is not relying on one central registry alone.
The core concept of our architecture is the *broker*. Instead of a central UDDI repository (Bellwood et al. 2002) each organizational unit is represented by its own broker. Our approach consists of a *set* of uniform brokers which communicate over a peer-to-peer network.
The broker combines all three activities, i.e. it invokes services, looks for available services, and registers the published services. With the help of a broker the organizational unit acts as a service provider and publishes its available services.
After its registration a broker knows all services of a local organization and can provide information about its local services. Each broker behaves in the same way and makes the lookup and invocation of the service possible – independent of how the represented organizational unit performs the service. It does not matter if the organizational unit has automated business process support (e.g. a workflow management system) or is fully paper based. The broker abstracts from the local process execution and provides a uniform interface for invocation of services for other parties.

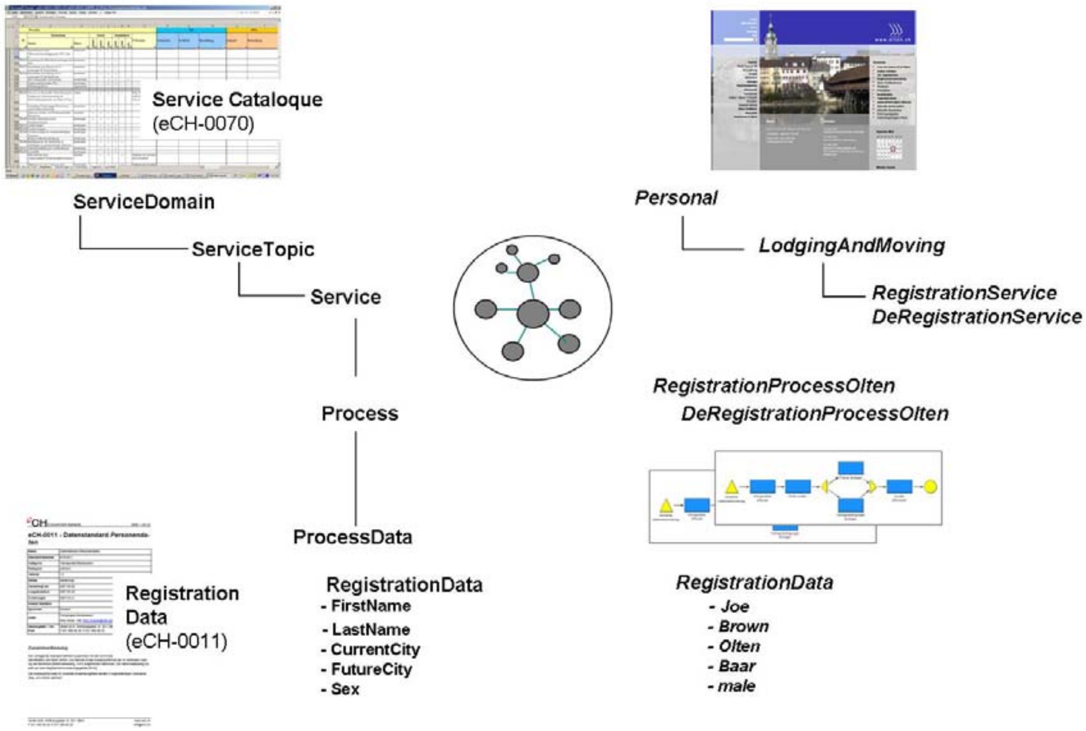


**Figure 4: Service classification and standard data**

The uniform service invocation interface also makes the broker attractive for the local organization itself. Because the broker provides a local point where also the local organization can start their own processes, it simplifies the enacting of new processes even for the local organization. Because the broker is under the complete control of the local organization it can be integrated into its IT support and maintenance processes. Therefore, we suppose the situation happening for central service repositories will not happen for the local broker and it will be maintained and supported by the local organization.

However, the brokers are also connected through a peer-to-peer network, where they can exchange information about their services. Therefore, each broker can also look up descriptions of other services which are offered by other local organizational units. A client does not need to know all brokers but only its local organization. It can get information about any service provided by the network. Instead of sending a service lookup request to a central repository a client sends this request to its local broker and still gets all available services including those offered by other brokers. The invocation of a service works in the same way, i.e. a local broker receives the service invocation request and forwards the request to the respective local brokers which provide these services. If a client wants to invoke a specific service of a specific organizational unit it always has to send the service request via its own broker representing its local organizational unit.

For a service invocation the concrete parameters like its URL or the method parameters of the service are not needed; instead, an ontology, defined in next section, is used to determine these services. Each service is described with respect to the ontology. Thanks to their ontology-based description the network of brokers is able to lookup and invoke the appropriate services with respect to their meaning.

As result of a request, all matching services in the network are invoked. This is a major difference to existing approaches which only try to find one answer to a service request. For example, when a person moves from one municipality to another, services for registration at the destination municipality will be requested. Depending on the local registration procedure of the new municipality the registration of a pet will be performed if the appropriate service is registered at the local broker of this municipality. If there are further related services available (e.g. notifying the school authority), then these services are also invoked.. The details of the request distribution and invocation process are described later.

While the broker network is completely distributed, brokers need a common vocabulary to perform the service lookup. In our case, all partners commit to the shared ontology which is central for the lookup and invocation of services. This ontology is discussed in the following section.

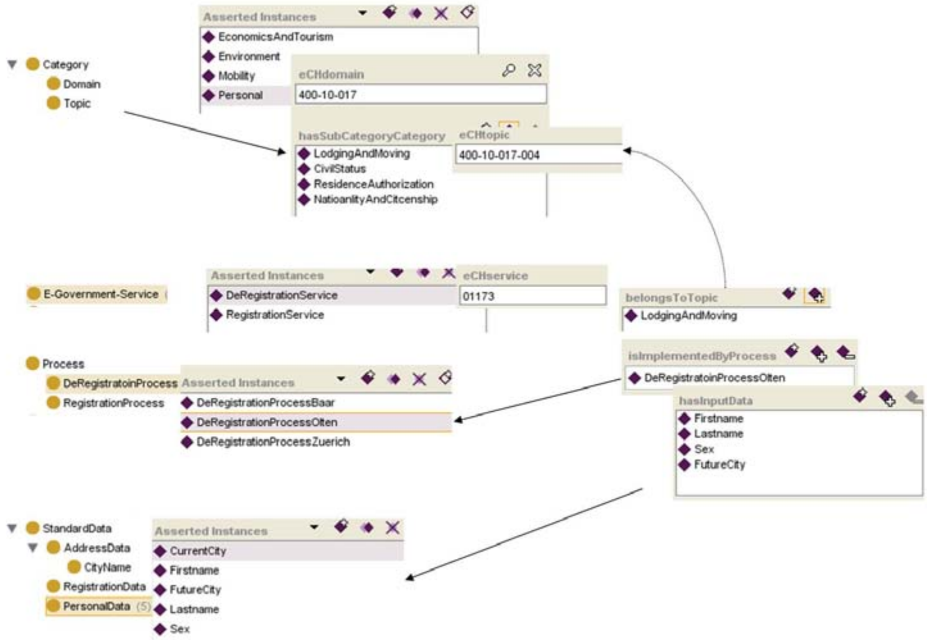


**Figure 5: Representation of service classification and standard data in Protégé**

## Ontologies for service description

For service description a small ontology is used[8]. Beneath the broker software this ontology is the only centrally maintained component of the introduced approach. The ontology is called 'E-Gov broker ontology' as it is queried by the broker for service detection. The content of the ontology is based on eCH-standards[9] and therefore already negotiated and well understood.

Figure 4 depicts the very little information needed for service description:

- service classification:
  a service is identified by an unique number, allocated to one and only one topic which belongs to a service domain. For example: the service 'registration' has the service no. 01172, the service 'de-registration' has the no. 01173; both are allocated to the topic 'lodging and moving' that belongs to the 'personal' domain.
- service data:
  mandatory data for service invocation and execution. For example: to invoke the service 'de-registration' the city a person wants to move from must be the city the municipality is responsible for. To execute the service, data like 'first name', 'last name', 'street number' etc. must be provided.

For further use the E-Gov broker ontology can be imported or included in a local ontology, e.g. in order to describe internal processes in more details.

Figure 4 gives an example of classification and standard data for the (de-)registration service. On the left hand side the hierarchical structure of the service catalogue is presented, whereas on the right hand side an example of a specific service is given. The lower part of the figure shows a cut out of the data for (de-)registration (meta data and data of an instance of the specific service).

The same example is modelled in OWL as depicted in Figure 5. The uppermost concept on the left hand side is 'Category' with its sub-concepts 'Domain' and 'Topic'. The highlighted instance of the 'Domain' concept is 'Personal' which is related to the 'Topic' instance 'LodgingAndMoving'. To this instance belong the terms 'DeRegistrationService' and 'RegistrationService' of the concept 'E-Government-Service. The 'E-Government-Service' concept has the property 'isImplementedByProcess'. In the example the 'DeRegistrationInstance' has three implementations, represented by the instances 'DeRegistrationProcessBaar', 'DeRegistrationProcessOlten' and 'DeRegistrationProcess-Zuerich'. With this simple construct, a distinct classification of any service implementation is possible.

Data, used as input (or output or during process execution) is represented by the concept 'StandardData' and its sub-concepts. Here the 'DeRegistrationProcessOlten' has input data 'Firstname', 'Lastname', 'Sex' and 'FutureCity' that are instances of the (sub-)concept 'PersonData'.

[8] It is a subset of the recently developed E-Gov Upper Ontology.

[9] mainly on the standards eCH-0011 for registration data and eCH-0070 for service classification.

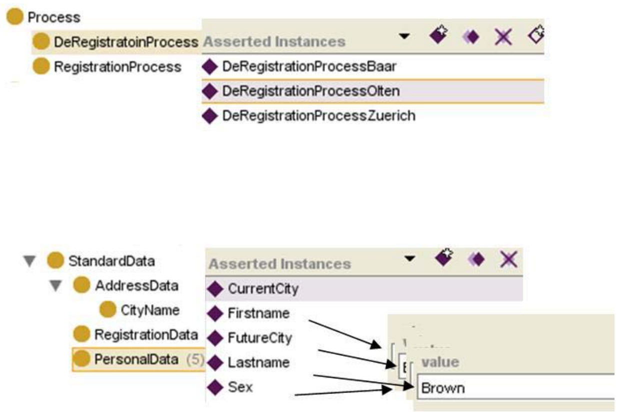


**Figure 6: Run Time instantiation**

The E-Gov broker ontology is very simple; instances of a service are modelled as instances of service execution. As the example shows for service *execution* instances of the instances of ‘PersonData’ (‘Olten’, ‘Joe’, ‘Baar’, ‘Brown’, ‘male’) are needed.

The chosen solution here is to model the requirement as a data property of the concepts and set the respective value of that property during run time as depicted in Figure 6. Even though this approach is successfully chosen in the FIT project[10] (Hinkelmann et. al. 2006) it is an open question how large data volumes can be handled as the ontology has to be loaded and updated during execution.
The introduced E-Gov broker ontology is used for service filtering (based on the classification concepts) and for decision making. Whether a service is invoked or not depends on the following criteria inferred from the ontology:

- a service is invoked when the condition(s) are met, e.g. the current city *is* the city the municipality is responsible for
- a service is executed even though mandatory data is missing e.g. for de-registration the street number is required but not provided; how incomplete data is handled must be defined locally
- a service is *not* invoked when data to decide on a condition is missing, e.g. the name of the current city is not provided.

As there is no top-down enforcement, Public Administrations have to be convinced, e.g. by pilot installations. To pave the path for that a continuous exchange between researchers of the University of Applied Sciences Northwestern Switzlerland and representatives of municipalities, cantons and state departments is maintained.

## Distributed Service Registry

As mentioned, the brokers form a peer-to-peer network to provide service lookup without requiring a central service registry. P2P networks organize their structure automatically and are fault-tolerant, meaning new brokers can easily join or leave the network.

The simplest way to implement a distributed registry would be to broadcast each service request to all participating brokers. Then, each broker would check, perform a local lookup and invoke all its matching services. Such an approach for service lookup has already been proposed for distributed service registries. (Siberski, Thaden, & Nejdl 2002) and describe a service registry network based on the JXTA P2P infrastructure. In this work, DAML-S is used to describes the available services. The network is unstructured and discovery requests are distributed in Gnutella style. A very similar approach is used by (Verma et al. 2005)

While this approach yields the correct result, it is inefficient and does not scale, because every broker would have to handle every single service request. To avoid the high broadcasting costs, it is advantageous to use index structures stored in a structured P2P network. We use a Distributed Hash Table (DHT) network (Stoica et al. 2001; Rowstron & Druschel 2001) as basic P2P infrastructure. A DHT offers the same functionality as a local hash table: it allows storing and retrieving arbitrary key/value pairs. All DHTs share the same basic concept. The peers in the network share a random hash function, i.e., a function which assigns pseudo-random numerical value to any given key. This hash function has a given numerical range. The range is partitioned between the peers, and each peer becomes responsible for one of these partitions. To add a new key/value pair to the network, a peer first computes the hash value for the key. Then, the key/value pair is forwarded to the peer responsible for the respective partition. Lookup is handled in the same fashion; the query key is transformed into a hash value which determines the responsible peer. This peer performs a local lookup and sends back the matching results. (Castro, Costa, & Rowstron 2005) have shown that DHT networks are not only more efficient, but also offer more reliable index access than unstructured variants. Proposals for DHT-based distributed service registries also do already exist. (Vu, Hauswirth, & Aberer 2005) use P-Grid for index storage. This network has the special feature that it considers quality-of-service criteria and is able to identify cheating peers, i.e., peers that do not keep the quality criteria they advertise. The SPiDeR network (Sahin et al. 2005) is based on the Chord DHT. It maintains

[10] FIT (Fostering self-adaptive e-government service Improvement using semantic Technologies) is a research project funded by the EU within the context of the Information Society Technologies (IST) programme (IST-27090, http://www.fit-project.org/).

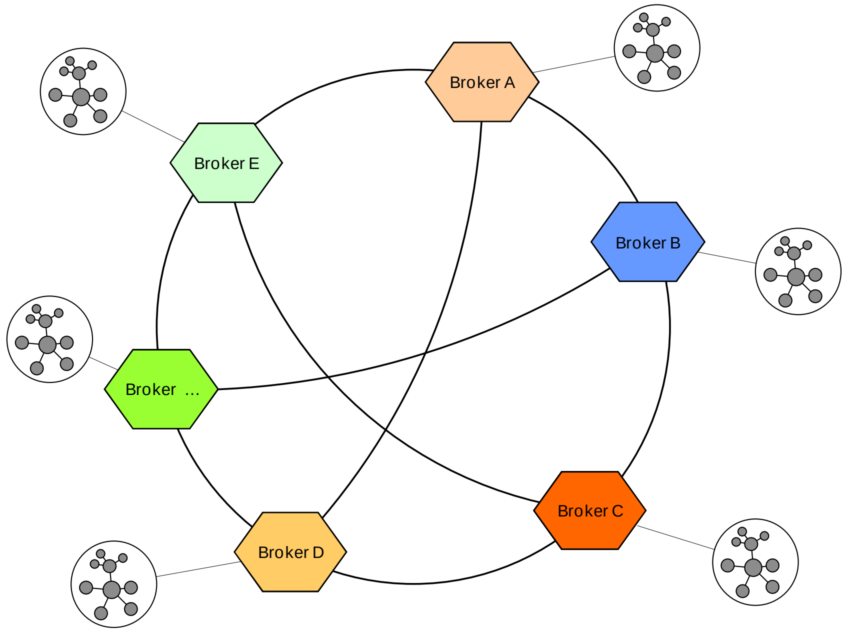


**Figure 7: CHORD Ring Network Topology**

keyword as well as ontology information about services in its index.

For contexts in which location is the dominant service selection criterion, the GWS registry network has been devised (Ma et al. 2005). It exhibits a topology representing an R+-Tree, where peers are positioned according to their geospatial location. This allows for very efficient area-based service discovery.
Another option is to structure the network according to ontology concepts. This is done in (Schlosser et al. 2002) and in (Sapkota et al. 2005).
(Gagnes et. al. 2006) give a good overview about requirements and selection criteria for registry networks.

To our knowledge, all of these approaches work in the same manner as traditional services, i.e., they return exactly one service which matches the request.

## Registry P2P Network Infrastructure

In the following, we illustrate our registry algorithm using CHORD (Stoica et al. 2001). In this DHT, peers are logically assigned a position in a ring, according to the hash value of their IP address. As in all DHTs, each peer gets assigned a range of hash values for which it becomes responsible. Figure 7 shows that a peer does not only maintain connections to its predecessor and successor, but also forms shortcuts to other peers in the network. These shortcuts connect the peers in such a way that for a network of size *n*, it takes not more than *log n* hops to reach any other peer. In our context, each service broker becomes a peer in the DHT network and thus contributes to the availability and reliability of the distributed index.

## Service Publication

The DHT needs to provide sufficient information to look up requested services. As we have described above, service requests are specified based on a shared ontology. A request specification always includes a category specifying the service type and a location where this service has to be performed. For example, the registration of a citizen in the location of Olten is the core of a request. We only use these two attributes to index the services. All brokers in the network add index entries for their offered services. Every services, is put into the DHT using its location and type attributes as keys.

But it is not sufficient to index the services only by the concepts directly referenced in the service description. The reason is, that DHTs only provide exact key lookup and do not support distributed reasoning; if we only create index entries for concepts appearing in the service description, the service discovery will fail for requests using related (e.g., more general), but not identical concepts. The discovery process therefore requires transformations of the request on the service client side, as well as derivation of additional index entries ont the service provider side. We will discuss the client-side request transformation in the next subsection. On the provider side, for both category and location we infer all concept memberships. For example, if the service location is the municipality Olten, we can infer the additional location canton Solothurn. For such a service, we add ‘Solothurn’ as key to the DHT as well.This step is especially important in cases where the E-

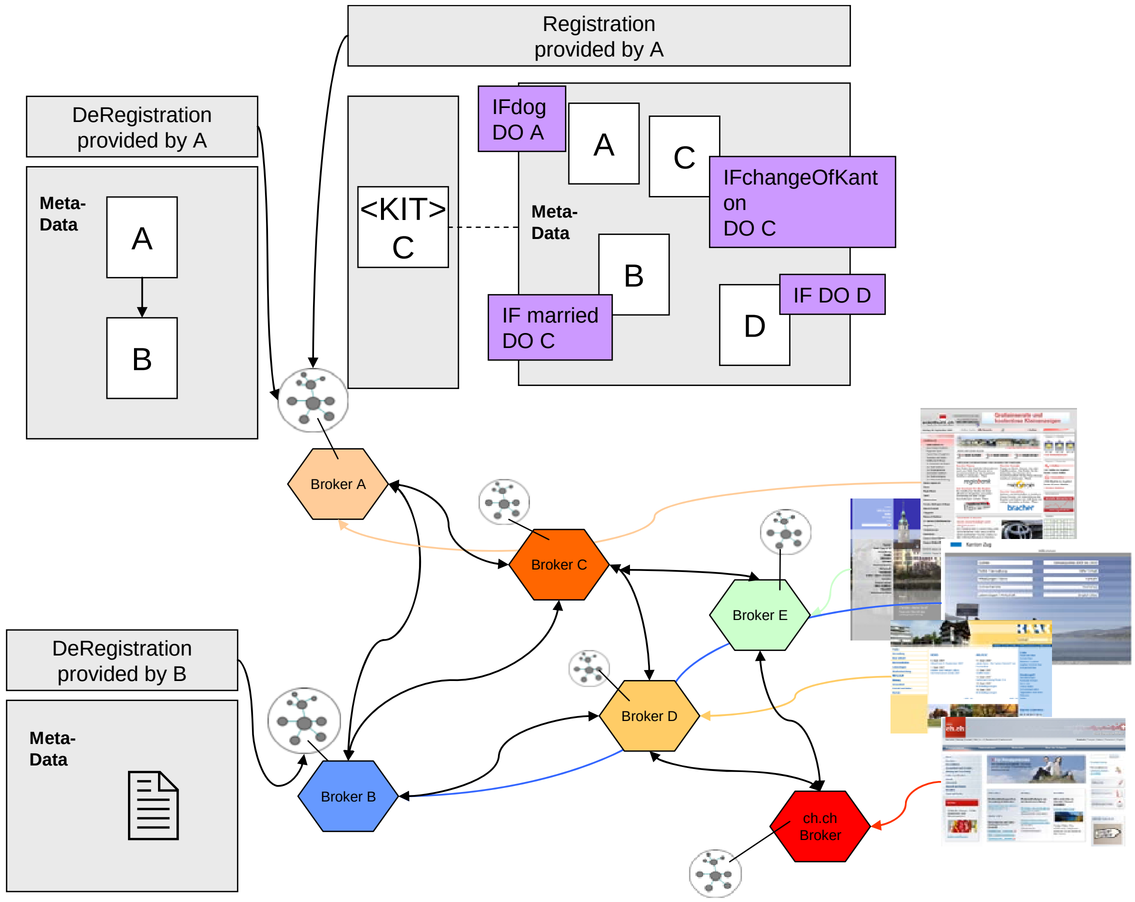


**Figure 8: One-Stop E-Government Service Announcement of Move**

Gov broker ontology has been locally extended, because it ensures that all services are indexed by shared concepts. Precomputing entailments is a technique also used to enhance efficiency of reasoning engines (Stuckenschmidt & Broekstra 2005). As the number of services offered by a broker is limited and services change rarely, the fact that some precomputation is required for the creation of index entries poses no problem with respect to efficiency.

## Service Discovery

The service matchmaking process works in two steps: The requesting broker first identifies matching candidates by looking up related locations, concepts and keywords in the DHT (*service lookup*). Result of this DHT lookup is a list of service ids (URLs) which are potential matches. For these candidates, the requesting broker performs a second step: it sends the original request to all brokers responsible for one of the identified services. These brokers perform an additional local matching (*service selection*). We now describe the discovery process in more detail.

**Service Lookup** First the keys for DHT lookup, i.e. the service category and location, have to be extracted from the service request. As with publication, we need to derive additional keys to cope with the limited lookup capabilities of the DHT. Based on the shared ontology the broker where the request originated determines which other concepts are entailed by the requested category and location. All resulting keys are used to lookup candidate services from the DHT.

**Service Selection** Not all services found during lookup do really match. Therefore, in the service selection step the service request is sent to all brokers providing service candidates, to perform a semantic matching locally.

The local service selection is needed because the service lookup via DHT only takes category and location into account to identify the set of brokers holding candidate services. The final set of matching services is therefore determined locally by each broker receiving the request.

Usually, service lookup will yield only few matches, and often just the requested services. Thus, an efficient retrieval of service candidates is ensured.

If several services fulfil the given constraints, all of them are invoked.

### Service invocation

Service invocation is straightforward. Each discovered service is invoked by the broker using the parameters provided as part of the service request. The services are designed to tolerate missing parameters. If such a situation occurs, a service call will trigger a new process on the target broker. This process initiates the retrieval of missing values out-of-band, e.g., by sending out a form via mail. Later on these values have to be added manually to actually perform the respective task.

## Service Implementation in Switzerland

The implementation of the introduced approach is exemplified by the use case 'announcement of move'.

Figure 8 depicts a citizen who can use any E-Government portal as entrance point for her request. To start the service she selects for example the appropriate form, the 'De-Registration-Form' on the portal of the municipality where she currently lives (Olten in the example). She enters the requested data, i.e. her name, the name of the city she currently lives in the name of the city she wants to move to etc. and submits the form.

Assuming that Olten has an E-Gov broker set-up, the broker is activated after submitting the data. It checks whether an internal process has to be invoked. In the example a process of the topic LodgingAndMoving is requested and the provided input data are 'firstName=Joe', 'lastName=Brown', 'sex=female', 'currentCity=Olten', 'futureCity=Baar', 'nationalityNotSuisse=true' and 'DeRegistrationRequested=yes'. If the broker detects processes related to the specified topic input, data is checked whether it meets the requirements of the service.

Assuming that the deregistration service can be started, the municipality of Olten is performing the process. During the execution in a separate process step, other municipality – e.g. Baar – has to be informed about the registration and the appropriate services have to be invoked. Without the broker architecture, normally a letter or an e-mail would be sent to someone in the next municipality; with the broker architecture, only the appropriate service request has to be launched for the local broker. The broker checks its local index structures; determine the superset of relevant brokers which should be informed by the service request, and forward the request for further checks to these candidate brokers. Obviously, the broker of the municipality of Baar is checked and the registration service is started. Because Joe Brown is a foreigner, the broker of the cantons Solothurn and Zug are additionally informed about the moving of Joe Brown. Their brokers also start corresponding services which are published for the (de)registration in the ontology.

## Conclusion and Future Work

One-stop E-Government services needs cross-organisation processes. But current approaches with a central repository for service registration lack support from local organizational units. In this paper a broker architecture is proposed which can invoke semantically enriched services without any global service registration. A set of brokers – for each municipality a seperat broker – are connected in a peer-to-peer network and broadcast service requests to appropriate destinations. It's the hope that the local brokers can be integrated in the local ICT maintenance and support process and therefore avoids the observed problems of a central registrys – the lack of support by local municipalities.

However the broker architecture can be extended in divers directions. A major problem of this architecture is the collection and distribution of all required data for the whole one-stop E-Gov process. In the example of the paper if the insurance number is not required for the deregistration process in one municipality, how can the required number be acquired for the following registration process. Because of the declarative service description, the whole set of data which might be used during the whole process may be determined. A dynamic form generator may then collect all the data in advance before the service is started. But how this can be done in detail is still an open issue.

Another interesting topic is the dropping of the one core ontology requirement. Instead of using one ontology each broker may use its own ontology. Then the problem of ontology mapping arises. Together with the dynamic peer-to-peer network this is really a challenge because the mapping has tio be done 'on demand'. It is absolutely unclear if such ad-hoc mapping is able to produce reliable results.

## References

Müller, W. 2005. *eGovCH - the Swiss eGovernment Architecture*. URL: http://www.isb.admin.ch/themen/egovernment/00069/ (10/04/2007)

Bellwood, T.; Celment, L.; Ehnebuske, D.; Hately, A.; Hondo, M.; Husband, Y. L.; Januszewski, K.; Lee, S.; McKee, B.; Munter, J.; and von Riegen, C. 2002. The UDDI specification V3. Technical report, Organization for the Advancement of Structured Information Standards (OASIS). available at http://uddi.org/pubs/uddi-v3.00-published-20020719.htm.

Castro, M.; Costa, M.; and Rowstron, A. 2005. Debunking some myths about structured and unstructured overlays. In *Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI'05).*

Engelmore, R., and Morgan, A. eds. 1986. *Blackboard Systems.* Reading, Mass.: Addison-Wesley.

Gagnes, T.; Plagemann, T.; and Munthe-Kaas, E. 2006. A conceptual service discovery architecture for semantic web services in dynamic environments. In *Proceedings of International Workshop on Semantics Enabled Networks and Services (SeNS'06).*

Hinkelmann, K., Probst, F., Thönssen, B. 2006. Agile Process Management Framework and Methodology. *AAAI Spring Symposium on Semantic Web Meets e-Government,* Stanford University, March 2006

Informatikstrategieorgan Bund 2007. *E-Government Strategie Schweiz*. URL: http://www.isb.admin.ch/themen/egovernment/00067/ (10/04/2007)

Ma, X.; Xie, K.; Liu, C.; and Li, C. 2005. Dynamic geospatial web services composition in peer-to-peer networks. In *Proceedings of 10th Asian Computing Science Conference (ASIAN)*, 257-258.

Rowstron, A. I. T., and Druschel, P. 2001. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Proceedings of Middleware 2001, IFIP/ACM International Con ference on Distributed Systems Platforms*, 329-350.

Sahin, O. D.; Gerede, C. E.; Agrawal, D.; Abbadi, A. E.; Ibarra, O. H.; and Su, J. 2005. SPiDeR: P2P-based web service discovery. In *Proceedings of Third International Conference on Service-Oriented Computing (ICSOC)*, 157-169.

Sapkota, B.; Vasiliu, L.; Toma, I.; Roman, D.; and Bussler, C. 2005. Peer-to-peer technology usage in web service discovery and matchmaking. In *Proceedings of 6th International Conference on Web Information Systems Engineering (WISE)*, 418-425.

Schlosser, M. T.; Sintek, M.; Decker, S.; and Nejdl, W. 2002. A scalable and ontology-based P2P infrastructure for semantic web services. In *Proceedings of the 2nd International Conference on Peer-to-Peer Computing (P2P)*, 104-111.

Siberski, W.; Thaden, U.; and Nejdl, W. 2002. A semantic web based peer-to-peer service registry network. Technical report, L3S Research Center. available at http://projekte.learninglab.uni-hannover.de/pub/bscw.cgi/d17999/A%20Semantic%20Web%20to-Peer%20Service%20Registry%20Network.pdf.

Stoica, I.; Morris, R.; Karger, D.; Kaashoek, M. F.; and Balakrishnan, H. 2001. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 Conference on applications, technologies, architectures, and protocols for computer communications*.

Stuckenschmidt, H., and Broekstra, J. 2005. Time-space trade-offs in scaling up RDF schema reasoning. In *Proceedings of the Workshop on Scalable Semantic Web Knowledge Base Systems*, 172-181.

Rudi Studer, R., Grimm, S., Abecker, A.2007. *Semantic Web Services: Concepts, Technologies, and Applications*. Springer Verlag, 406 pages

Verma, K.; Sivashanmugam, K.; Sheth, A.; Patil, A.; Oundhakar, S.; and Miller, J. 2005. METEOR-S WSDI: A scalable P2P infrastructure of registries for semantic publication and discovery of web services. *Information Technology and Management* 6(1):17{39}.

Vu, L.-H.; Hauswirth, M.; and Aberer, K. 2005. Towards P2P-based semantic web service discovery with QOS support. In *Proceedings of the Workshop on Business Processes and Services*, 18-31.