AMCIS 2010 Proceedings

Americas Conference on Information Systems (AMCIS)

8-2010

RFID in the Cloud: A Service for High-Speed Data Access in Distributed Value Chains

Holger Ziekow
Humboldt-Universität zu Berlin, ziekow@wiwi.hu-berlin.de

Benjamin Fabian
Humboldt-Universität zu Berlin, bfabian@wiwi.hu-berlin.de

Cristian Müller
Humboldt-Universität zu Berlin, cristian.mueller@wiwi.hu-berlin.de

Oliver Günther
Humboldt-Universität zu Berlin, guenther@wiwi.hu-berlin.de

Recommended Citation

Ziekow, Holger; Fabian, Benjamin; Müller, Cristian; and Günther, Oliver, "RFID in the Cloud: A Service for High-Speed Data Access in Distributed Value Chains" (2010). *AMCIS 2010 Proceedings*. 256.
http://aisel.aisnet.org/amcis2010/256

RFID in the Cloud: A Service for High-Speed Data Access in Distributed Value Chains

Holger Ziekow
Institute of Information Systems
Humboldt-Universität zu Berlin
ziekow@wiwi.hu-berlin.de

Benjamin Fabian
Institute of Information Systems
Humboldt-Universität zu Berlin
bfabian@wiwi.hu-berlin.de

Cristian Müller
Institute of Information Systems
Humboldt-Universität zu Berlin
cristian.mueller@wiwi.hu-berlin.de

Oliver Günther
Institute of Information Systems
Humboldt-Universität zu Berlin
guenther@wiwi.hu-berlin.de

ABSTRACT

Radio-Frequency Identification (RFID) is emerging as an important technology for exchanging information about physical objects along distributed value chains. The influential standardization organization EPCglobal has released standards for RFID-based data exchange that follow the data-on-network paradigm. Here, the business-relevant object data is provided by network services, whereas RFID tags are only used to carry a reference number for data retrieval via the Internet. However, as we show in this paper, this paradigm can result in long response times for data access. We present experiments that explore what factors impact the response times and identify obstacles in current architectures. Based on these analyses, we designed a cloud-based service that realizes high-speed data access for data-on-network solutions. We further present simulation experiments analyzing the benefits of our cloud-based concept with regards to fast RFID-data access and reduced infrastructure cost through scale effects.

Keywords

RFID, EPCglobal Network, EPCIS, Cloud Computing, Content Distribution Networks.

INTRODUCTION

Radio-Frequency Identification (RFID) is an important technology for increasing transparency and efficiency in value chains. Physical objects can be equipped with simple chips – called tags – which communicate with RFID readers via radio waves. The most common tags in logistic applications are in general only capable of storing little data (e.g., identification numbers) and can only process simple operations. The "intelligence," the decision making, and product-information storage and retrieval will not happen at the tags, but at the back-end, a middleware or application layer, and also via the Internet.

The paradigm behind this design is known as *data-on-network* (Diekmann et al, 2007) and facilitates distributed storage of possibly large amounts of data about the objects at hand, increases reliability of information access in case of insecure environments and transportation paths where tags may get lost, and, not least, allows to use cheap tags of low computational intelligence. This paradigm is supported by the influential industry consortium EPCglobal that provides a standardization of a global identification scheme for objects, known as Electronic Product Code (EPC). This EPC, which is globally unique, can be used as a reference number to retrieve information from the EPCglobal Network (EPCglobal, 2007), a widely distributed system of databases, the EPC Information Services (EPCIS).

In this paper, we investigate the shortcomings of this EPCglobal architecture design with respect to process control in distributed value chains under real-time constraints. In such applications, downstream business operations may require fast access to data records from previous steps in the supply or processing chain. These data records will generally be stored on remote locations, possibly distributed around the globe and with long access latencies. Long access latencies can cause problems wherever a process requires fast system responses, e.g., to make a routing decision on a conveyor. We present a solution for the problem of access delays that uses concepts of cloud computing. Our solution is specially designed for applications in distributed value chains, and is compatible with existing standards and completely transparent to the information clients. The design is based on physically distributed data centers that mirror RFID data from different value-chain stations. With this architecture we (1) reduce request concurrency by distributing load, and (2) reduce network delay by providing RFID data via short network paths. The presented solution aims at improving object related information exchange in distributed value chains.

Our paper is structured as follows. In Section 2, we describe how data is accessed in the EPCglobal Network, and motivate why access speed is critical in specific applications. Section 3 investigates limiting factors for access speed, based on earlier experimental work on PlanetLab. In Section 4, we present a new architecture, the *EPCIS Cloud*, to ensure fast RFID-data access based on the pooling of EPCIS servers. Simulation experiments presented in Section 5 show some major advantages of this architecture. Related work will be discussed in Section 6.

ACCESSING RFID DATA IN THE EPCGLOBAL NETWORK

According to the influential standards of EPCglobal, information about an object should in general not be stored on its RFID tag itself, but instead be supplied by distributed servers on the Internet. The resulting EPCglobal Network (EPCglobal, 2007) promises to let many parties – manufacturers, suppliers, shops, or after-sale service providers – dynamically register EPC Information Services (EPCIS) for the objects they are concerned with, resulting in a flexible way to exchange product related information. Using the Electronic Product Code as a look-up key, supporting infrastructure services like the Object Naming Service (ONS) (EPCglobal 2008) and EPCIS Discovery Services (BRIDGE 2009) will facilitate to locate the relevant EPCIS servers for an object (Figure 1, Step 1). Those information servers are then contacted directly (Steps 2, 3). This procedure aims to improve the information flow between companies while objects are transferred from suppliers to manufacturers, distributors, retail stores, and customers, and facilitates cooperation and efficiency within supply chains.

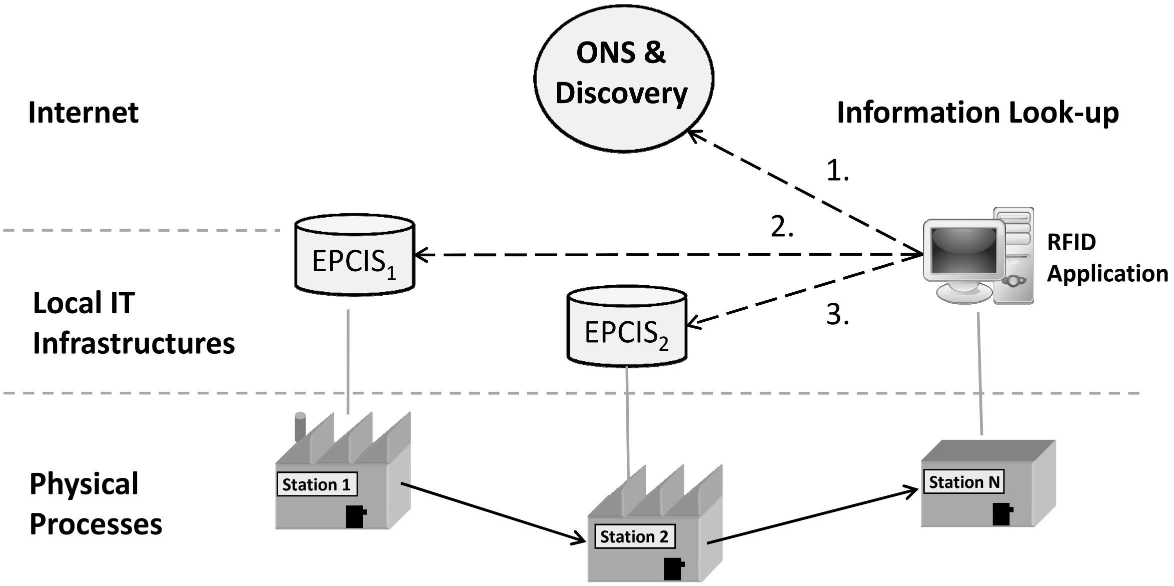

Figure 1. Accessing RFID Data in the EPCglobal Network

The main question we raise in this paper is if this architecture design supports applications under real-time constraints. Downstream business operations may require fast access to data records from previous steps in the supply or processing chain. For example, a manufacturer may route items through its production lines based on object-related information. Also, companies may quickly check some object-related data at the material intake to reject invalid deliveries right away. Here it is crucial that the process of accessing necessary information does not slow down operations. Manual operations may usually require system responses faster than 0.5 seconds to not delay the process (Ivantysynova and Ziekow, 2008). This threshold could be even lower for automatic processes. Meeting real-time requirements is particularly challenging if the application needs RFID data from very remote locations.

LIMITING FACTORS FOR FAST DATA ACCESS

In this section we present experiments that show how different factors influence the responsiveness of an EPCIS server. We identified two factors: the load of a node, and the distance between the sender and the receiver. The experiments explicitly do not consider the discovery of previously unknown EPCIS. We chose this abstraction because we aim to optimize the environment for accessing EPCIS – optimizations of the EPCIS-discovery phase may not be necessary in every scenario. The discovery phase corresponds to Step *1. Information Lookup* in Figure 1, and is optional in cases where the EPCIS web address (URL) is known in advance by pre-configuration or caching.

Application-Level Network Delay in RFID Applications

As part of our work we conducted experiments on the application-level network delay for accessing RFID data (Ziekow, Fabian and Müller, 2009). For these experiments, we deployed EPCIS on globally distributed locations using the PlanetLab (PlanetLab 2009). We deployed EPCIS on servers in USA, Germany, China, and Japan. All regions participated with three to five servers – this variance in server numbers is due to unreliable servers on PlanetLab, an effect offering real-world conditions. As EPCIS software we used the open source implementation provided by Fosstrak (Fosstrak 2009).

In our experiment, we issued queries between the servers and recorded their response times. For a period of three days, we let every server issue one query per minute to the EPCIS on all other servers. We used a very simple EPCIS query that retrieves all data records about a single object (or EPC). We chose these very simple requests to measure delays under rather ideal conditions. Consequently, our experiments show results along the lower bound for delays. More complex queries are likely to yield even longer response times.

Our work showed that the physical distribution of data sources significantly impacts the response time (Ziekow et al., 2009). Results of this work are visualized in Figure 2 (left). It shows the minimal query response times in dependence of the physical distance. Visualized are combined results for queries to Boston, Berlin, Tokyo, and Osaka, issued from all regions. Parts of the delay may be explained due to poor performance of some servers that concurrently run experiments of other research groups in the PlanetLab infrastructure. Despite this distortion, our experiments clearly show the impact of network delay over the distance. The correlation between physical distance and response time is due to the increased network delay for long distance communication. Note, that network delay is independent of the used server hardware and inherent in distributed data access. This effect is therefore inherent in accessing distributed RFID data.

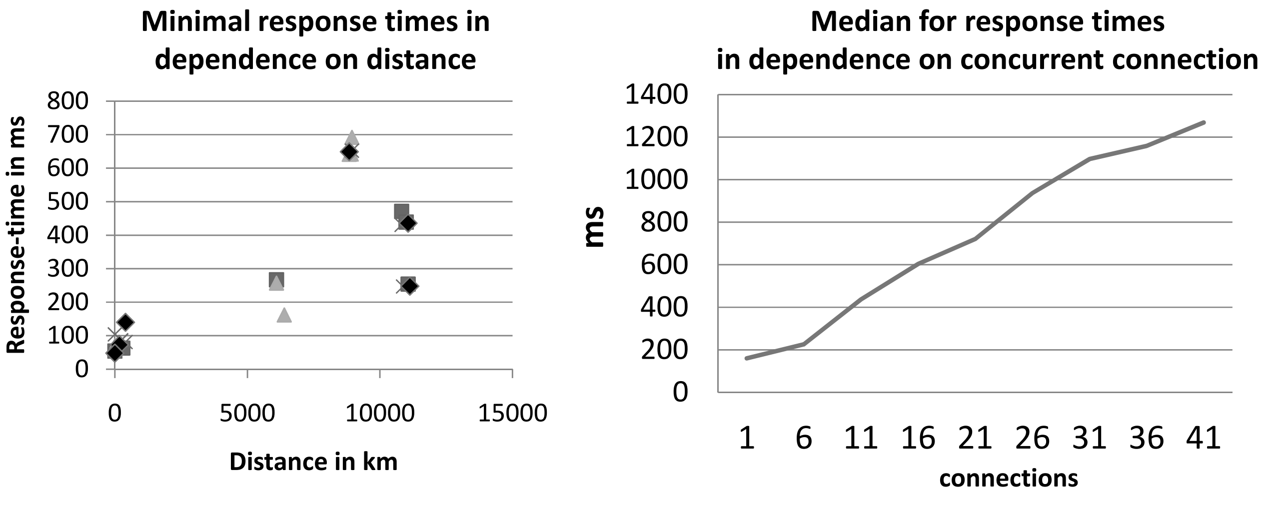

Figure 2. Minimal response times in dependence on physical distance (left) (see Ziekow et al., 2009) and response times of a dedicated EPCIS Server for concurrent queries (right)

Our experiments showed that the physical distribution of EPCIS servers accounts for a significant proportion in query response times. This effect is of particular importance to applications in globally distributed value chains..Given our performance measurements, it seems currently unlikely that EPCIS-based solutions can support time-critical processes in distributed value chains. High performing servers may solve parts of the problem. However, network delay is inherent in accessing remote data sources and our experiments showed the significance of this effect in accessing remote EPCIS.

Concurrency at RFID Data Sources

For this experiment we inducted a certain amount of stress into an EPCIS server and measured the response times. The EPCIS node was dedicated to its assignment and only served EPCIS queries. Again we used Fosstrak (Fosstrak 2009) as EPCIS software. The dedicated Server was a Pentium III with 512MB of RAM. To measure the effect of the stress we set up a dedicated client. The client – from now on also referred to as measure point - recorded the response time for a sequence of 10000 consecutive queries. To generate additional load on the server, we recruited a number of nodes from PlanetLab – referred to as stressors. These stressors consecutively issued the same query as the measure point as fast as possible. This procedure was repeated with an increasing amount of stressors. The impact on the response time is depicted in Figure 2 (right). The Figure shows the median of the measured values for certain numbers of stressors. The graph shows a clear reaction to the increasing load. Note that the measure point and the stressors used the same simple query as in the first experiment. More complex queries are likely to increase the response time. Note that the exact relation between load and delay depends on the used hardware and software. However, we believe it is fair to assume that different technical setups would show similar effects.

DESIGN OF A CLOUD SERVICE FOR RFID DATA

In this section we present the design of a cloud based service that facilitates fast access to RFID-data. It addresses the delay factors that we identified in our analysis in Section 3. Our solution design is driven by two main design goals. The foremost goal is to accelerate network based RFID-data access. The second goal is staying conform with existing standards where possible. Apart from increased access speed it should be opaque to the users if our solution is used or not. We strongly believe that this is of major importance for successful implementation. In the following we provide details about our solution.

Architecture

Our experiments in Section 3 showed that we must consider two technical dimensions to accelerate RFID-data access. One dimension is the network delay and the other is concurrency of requests. To reduce network delay we must proactively place RFID data close to the origins of requests. To handle concurrent requests we must distribute the load across multiple processing resources. These basic considerations lead to a design where a distributed set of servers is used for providing RFID data. We refer to single servers in this set as Cloud-EPCIS and the whole infrastructure as *EPCIS-Cloud Infrastructure*. An illustration of the basic architecture is given in Figure 3 (left). Cloud-EPCIS provide RFID data on behalf of the EPCIS at the corresponding value-chain location. For simplicity we refer to the EPCIS at value-chain locations as Master-EPCIS. RFID applications retrieve data from the EPCIS-Cloud Infrastructure instead of directly communicating with Master-EPCIS across the value chain. We thereby decouple resources for data provisioning from the originating value-chain locations.

This decoupling allows serving data requests from using nearby Cloud-EPCIS instead of potentially distant Master-EPCIS in the value chain. We can therefore resolve the problem of long network delays. The network delay for retrieving data from the EPCIS-Cloud Infrastructure depends on the distance to the next Cloud-EPCIS. The desired delay reduction is therefore achieved through adjusting the number of servers. Decoupling the data provisioning from the value-chain locations also allows tackling the problem of concurrent data requests. Moving RFID data to the cloud inherently distributes the request for participating value-chain locations. Again, one can achieve the desired delay reduction by adjusting the number of servers.

The EPCIS-Cloud infrastructure comprises three elements that together facilitate the distributed data provisioning. The first element is a set of Cloud-EPCIS. Cloud-EPCIS mirror RFID data from the Master-EPCIS of participating value-chain locations and provide EPCIS interfaces for RFID applications. The second element is a distribution mechanism. This mechanism is in charge of distributing the RFID data from Master-EPCIS to suitable Cloud-EPCIS. The third element is a redirection mechanism for RFID-data requests. This mechanism ensures that requests from RFID applications go to the closest Cloud-EPCIS instead of the Master-EPCIS in the value chain. An overview of the three elements and their interaction is given in Figure 3 (right).

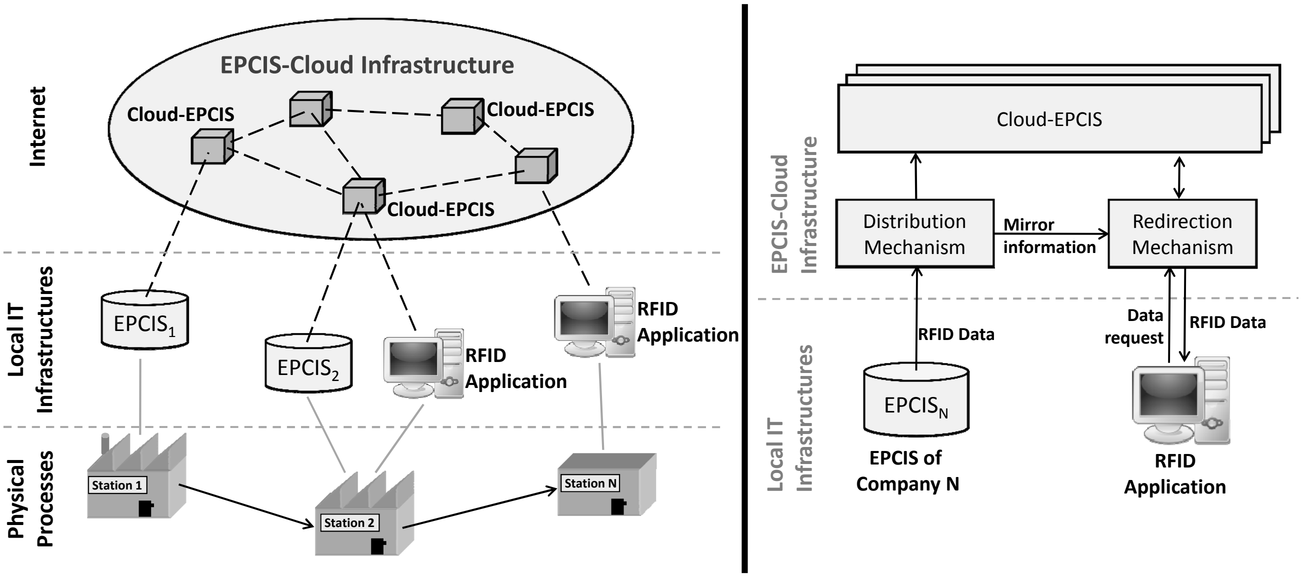

Figure 3. EPCIS-Cloud Architecture

Hosting Cloud-EPCIS

The primary task of Cloud-EPCIS is to mirror RFID data from Master-EPCIS. We suggest an implementation of this functionality solely based on EPCglobal standards. Writing data to the mirror is possible using the capturing interface of EPCIS. At this point however, it is possible to use proprietary interfaces and still being standard conform for any interaction with users of the Infrastructure. For retrieving mirrored RFID data, one can use the EPCIS query interface. Using this standard interface is crucial for making the EPCIS-Cloud Infrastructure transparent to users. RFID applications can thereby access data from the cloud as if they were directly accessing it from Master-EPCIS.

The key challenge in providing Cloud-EPCIS for a Master-EPCIS is to pick suitable servers. Our experiments in Section 3 show that physical location has a major impact on the delay in data access. The main goal for picking host servers is therefore to have Cloud-EPCIS as close to the accessing RFID applications as possible. Another goal is to distribute requests such that peak loads do not cause significant delay. The combination of both goals defines the optimization problem for picking servers. The default strategy to choose server locations evenly distributed across the globe. This is the only feasible strategy if nothing is known about the value-chains structure. More sophisticated strategies can be considered if some background knowledge is available. A detailed discussion of candidate placement strategies is beyond the scope of this paper. However, limiting server placements to certain regions is likely a good heuristic for many cases.

Embedding Cloud-EPCIS in the EPCglobal Architecture

The key to embed Cloud-EPCIS in the EPCglobal architecture is to reestablish a redirection mechanism that directs data requests to the closest Cloud-EPCIS. There are several options for implementing a redirection mechanism, as depicted in Figure 4. These options depend on the mechanisms how an EPC is resolved to EPCIS addresses in future generations of the EPCglobal Network.

The resolution path, as currently described in EPCglobal standards, involves transforming an EPC to a corresponding DNS name, the consultation of the ONS to retrieve a set of DNS names for the EPCIS, a consultation of the DNS to resolve those names into IP addresses, and a final contact to the EPCIS to download or update the actual RFID data for this EPC. Currently, mainly the manufacturer EPCIS can be located using ONS, but future discovery services, once finally specified, will offer a much more flexible selection of data sources. Proposals for P2P-ONS based on Distributed Hash tables (e.g., Fabian and Günther, 2007) could already today be deployed to resolve an EPC via its corresponding Overlay ID into a nearly arbitrary set of EPCIS servers (DNS name or IP Address), and could even deliver the actual data (Fabian, 2009).

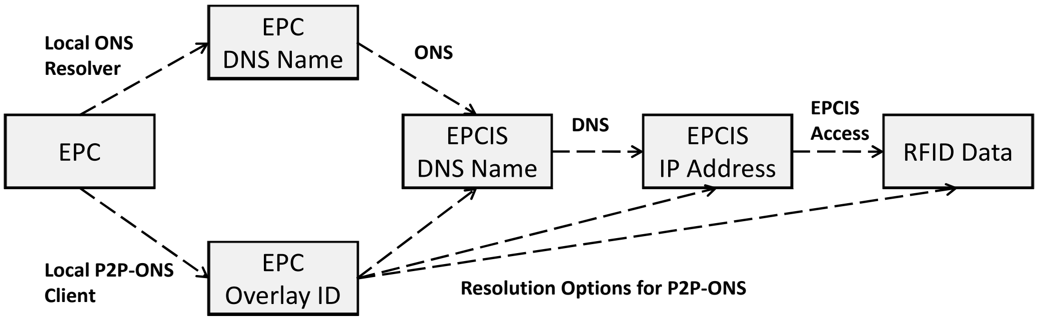

Figure 4. Alternative EPC-Resolution Paths

A comparison of some possible redirection mechanisms is presented in Table 1, for a detailed survey see Barbir et al. (2003). For our Cloud-EPCIS architecture proposed in this paper, we choose a direction mechanisms based on DNS, since this is, unlike Discovery Services, available at the current time, would not change EPCglobal standards like P2P-ONS, and offers a higher flexibility than the current – mostly manufacturer-focused – ONS standard. Further, it also works if no dedicated discovery phase is used in a specific application scenario (Figure 1, Step 1).

Using DNS for redirection would come into play, once ONS, P2P-ONS, or a Discovery Service of the EPCglobal Network have returned an EPCIS DNS name like epcis.example.com to the client. Subsequently, the client consults its DNS server for a resolution of this name to an IP address. Disregarding caching mechanisms in this example, the DNS server consults the DNS root and a DNS server for .com, to reach an authoritative name server for example.com.

Instead of simply returning the IP address of the Master EPCIS (which may not be the latency-optimal for the client), another redirection step to a DNS server of the EPCIS-Cloud Provider is conducted, e.g., to dns1.example.com.epcis-cloud.com. The provider of the EPCIS Cloud maintains a hierarchy of DNS to internally delegate the query based on the source IP address, until it reaches a DNS server that can provide the IP address of the Cloud-EPCIS that is assumed to be nearest to the client. This IP address is returned to the client, which then contacts this Cloud-EPCIS to retrieve data, without noticing that this is not the originally queried-for Master EPCIS. A similar DNS redirection mechanism appears to be used in the proprietary Akaimai CDN, as documented by Su et al. (2006)

Redirection Mechanism	**Changes**	**Redirection control by**	**Conformity to EPCglobal Standards?**	**Estimated Complexity of Configuration?**	**Reference and Notes**
IP Anycast	BGP routes	Internet Service Provider	Y	High	Abley & Lindqvist (2006)
DNS	ECPIS DNS name or IP address	EPCIS Provider	Y	Medium	Su et al. (2006)
ONS	ECPIS DNS name	EPC Manager	Y	High	EPCglobal (2008); Manufacturer EPCIS
Discovery Service	ECPIS DNS name	EPCIS Provider	(Y)	Low	BRIDGE (2009); not yet standardized
P2P-ONS	ECPIS DNS name or IP address	EPCIS Provider	N	Low	Fabain and Günther (2007)

Table 1. Comparison of Redirection Mechanisms

Distribution Mechanism

The distribution mechanism distributes the RFID data from Master-EPCIS to Cloud-EPCIS. It therefore registers a standing query at the Master-EPCIS and forwards new data to Cloud-EPCIS. The main design question concerns the particular forwarding mechanism. Yet, the performance requirements of the distribution mechanisms are relatively low. This is because in the targeted application domain we can afford relatively long delays in mirror updates. RFID applications at a value-chain location do not need high-speed access to RFID data before the corresponding object is shipped. Thus, updating Cloud-EPCIS must only be faster than the shipping; which can take hours, days or even weeks.

Theoretically, any content delivery method can be used, ranging from direct downloads to using P2P protocols. However, particularities of the application domain yield some desirable properties of the implementation. One requirement is that the distribution mechanism should scale well with the number of selected Cloud-EPCIS. This makes P2P technologies like the BitTorrent protocol suitable (BitTorrent 2009, Androutsellis-Theotokis & Spinellis 2004). Using such technologies, Cloud-EPCIS can download RFID data from other Cloud-EPCIS and thereby reduce the burden for the Master-EPCIS. Another requirement is that data distribution within the cloud should not slow down data access for RFID applications. We can achieve this by only distributing data during idle times of Cloud-EPCIS. We observed in simulations that requests at EPCIS come in bursts, leaving enough idle time to use for data distribution.

Business Model for Service Providers

Our solution is designed to reduce network delays for applications where fast data access is needed. Hosting such a service requires a globally distributed server infrastructure. It is infeasible (if not impossible) that each provider of RFID data runs the described distributed solutions. Instead, our service is well suited to extend the product portfolio of cloud providers. Specifically providers that offer content distribution services already possess the needed infrastructure.

A key concept in cloud computing is that cloud providers can use resources more efficiently through statistical multiplexing, and may operate at lower cost than medium-sized data centers (Armbrust et al. 2009). Our design promises to show this effect by pooling several EPCIS and thereby smoothing peak loads at RFID databases. Also our design promises to show scale effects when a growing number of Cloud-EPCIS is hosted by one provider. Section 5 analyzes these effects in detailed experiments. Together, statistical multiplexing and scale effects allow cloud providers to provide our service at lower cost than regular players in distributed value chains can do. It is therefore appealing for cloud providers to add our service to their product portfolio.

SIMULATION EXPERIMENTS

In this section we present results of simulations that analyze load distribution in the EPCIS-Cloud Infrastructure. With these experiments we investigate how cloud providers can exploit statistical multiplexing and scale effects when hosting our service. We analyzed these effects with discrete event simulation in the Omnet++ framework (OMNeT 2009).

We simulated a value chain comprising one supplier tier with ten suppliers, a consumer tier with then consumers, and a single distribution center (see Figure 5 right). We used a Kanban system for placing orders. A FIFO policy controlled the stock at the distribution center. We simulated the demand at customers using a random variable with a uniform distribution and maximal 10000 items per order, using two orders per day. The lead time for deliveries to the distribution center was governed by a normal distribution with expectation value of 1.5 days and a variance of six hours. For deliveries from the distribution center to consumers, we assumed a normal distributed lead-time with an expectation value of two days and a half-day variance.

With this setup we simulated requests for RFID data along the value chain. The simulation generated requests to EPCIS at the intakes of the distribution center and the consumer. We set the intake speed to one item per second and requested the corresponding RFID data from the Master-EPCIS of each preceding value-chain location. At each server we monitored the load. As a load measure we logged for each incoming request the number of concurrently handled requests upon arrival.

First we analyzed the effect of pooling resources for several value-chain stations in the cloud. We therefore varied the number of Master-EPCIS hosted at cloud data centers. To study the effect of pooling resources, we chose only one mirror for each Master-EPCIS. Figure 5 (left) shows results obtained by simulating four weeks of supply chain activity. The Y-axis denotes the peak load at the data center (measured as open connections when request arrives). The X-axis plots the number of Master-EPCIS hosted at the data center. The setting with only one Master-EPCIS resembles the original EPCglobal model without the EPCIS-Cloud-Infrastructure. Here, every value-chain location runs its own data center for providing RFID data. The settings with more than one Master-EPCIS use the EPCIS-Cloud-Infrastructure for pooling resources.

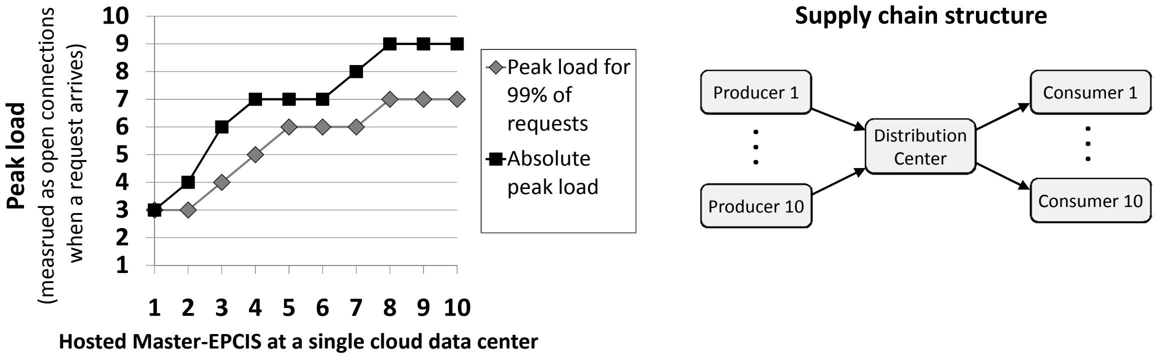

Figure 5. Peak load at cloud servers (left) for a simulated supply chain (right)

The results in Figure 5 (left) show that the peak load at a data center grows sublinearly with the number or hosted Master-EPCIS. Consequently, the EPCIS-Cloud-Infrastructure allows taking advantage of scale effects and thus save processing resources. For illustration consider an abstract processing resource *R* that allows timely response to one request at a time. A data center that hosts only one Master-EPCIS needs three such processing resources to answer all requests timely. Thus, all 10 producers together would need to provide 30*R* for fast data access. However, a shared data center in the EPCIS-Cloud-Infrastructure needs only 9*R* to ensure timely responses. The effect is even more drastic when only 99% timely responses are required. This quality of services requires 30*R* with one Master-EPCIS per data center and only 7*R* in the EPCIS-Cloud-Infrastructure.

In additonal experiments we analyzed the effect of load distribution among multiple Cloud-EPCIS. In the experiments we tested two scenarios. In scenario 1 we used a single data center for hosting the EPCIS of all ten producers. In scenario 2 we used two data centers C1 and C2 in the cloud. This could be for instance data centers on two different continents. We assumed that each data center is in proximity to five of the ten consumers. We further assumed that the distribution center is closer to data center C2. Using this setup we studied how the number of data centers in the cloud affects load distribution.

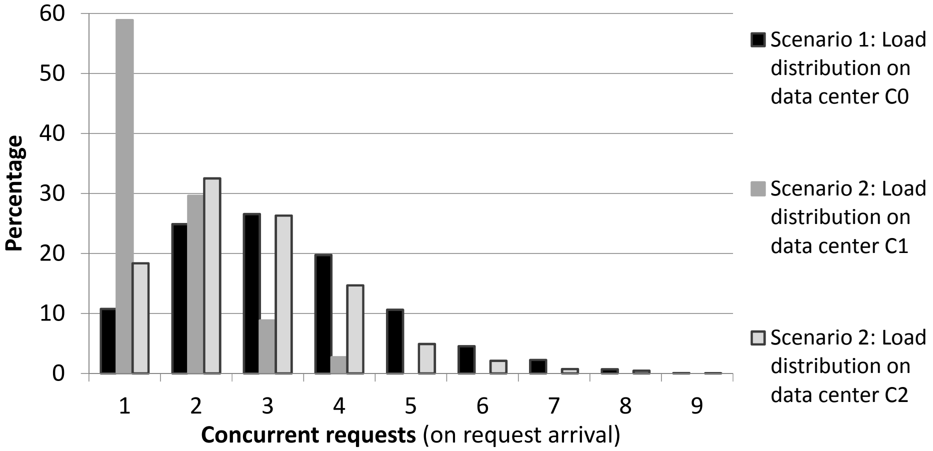

Figure 6. Load distribution at Cloud-EPCIS

Figure 6 displays the results of the simulations. The black columns show the load distribution for the single data center in scenario 1. The grayish columns show the load for the data centers C1 and C2 in scenario 2. The load distributions for C0 and C2 are very similar because they both served requests from the distribution center in the value chain. However, the load distribution for C0 has a longer tail because C0 serves more consumers than C2. C1 only serves requests from the five consumers in its proximity. It has a differently shaped load distribution with no peak loads higher than 4 concurrent requests. This has important implications on the required processing resources in the cloud. We found that running two data centers require less than twice the resources for one center. For illustration consider again the abstract processing resource *R* for timely handling one request. C0 required 9*R* to handle the peak loads. C1 requires 9*R* and C2 4*R*. Thus the two data centers in the cloud need together only 13*R* (instead of 18*R*) for timely providing all RFID data at two different locations. This finding is important for the scalability of the infrastructure.

In section 3.2 we have shown that concurrent request can significantly delay the data access. Each data center must provide enough resources to cope with the problem. Our simulation shows that load distribution in the EPCIS-Cloud Infrastructure reduces the corresponding burden per data center. On average, shared cloud data centers need fewer resources for timely handling peak loads than single data centers outside the cloud. Thus the infrastructure scales well with the number of data centers.

RELATED WORK

Our work is related to commercial Web content distribution networks like Akamai (Saroiu et al. 2002; Su et al. 2006), which are used today to reduce the load and response-time of very popular Web sites. Though the redirection mechanisms used by these networks may also work for EPCIS-Clouds (see Section 4.3), there is a fundamental difference in the assumed popularity of the mirrored content, which concerns Web content of interest to hundreds of thousands of Web users, and which can be easily cached. In contrast, when using EPCIS-Clouds for enhancing response-time in value chains, most of the time data would be only accessed a few times and by few users, mainly correlated to the steps of the value chain involved. Therefore, the content-distribution algorithms and server-location strategies will look considerably different.

Earlier work on scalability in EPC based data access was conducted by Williams et al. (2008). Unlike in our work they focused on resulting traffic volumes rather than on delay in data access. Related work on scalable distributed storage systems like P2P systems (see Androutsellis-Theotokis & Spinellis 2004 for a survey) could be highly relevant in future research on choosing an optimal data distribution technology for EPCIS Clouds. However, enhancing throughput is not a primary design goal of EPCIS Clouds, but minimizing the delay, which may constitute an important exclusion criterion for some P2P systems in this application.

CONCLUSIONS

Providing RFID data via network services is of vital importance for inter-enterprise RFID applications. In our experiments we analyzed the impact of network delay and request concurrency on data retrieval with RFID specific standards. The results show that the data-on-network paradigm poses serious challenges for ensuring fast response times. Thus, applications that need high-speed access to RFID data require solutions for accelerating data retrieval.

Based on this insight we designed a service concept that allows cloud providers to speed-up network-based RFID data retrieval. Our solution is specially designed for applications in distributed value chains. It is compatible with existing standards and completely transparent to the user. The proposed architecture (1) reduces request concurrency by distributing load, and (2) reduces network delay by providing RFID data via short network paths. The amount of network-delay reduction depends on the number of used data centers and on how the system users are geographically distributed. A limitation of the architecture is that achieving zero networks is infeasible because it would require a Cloud-EPCIS at every value-chain location. Finding a good compromise between the number of Cloud-EPCIS and reduced network delay is subject to future work.

We have shown in simulations that our architecture scales well with the number of Cloud-EPCIS. That is, the required processing resources per Cloud-EPCIS and data center grow sublinearly with the number of data centers. Ideally, the Cloud-EPCIS infrastructure hosts a Cloud-EPCIS close to every user. Options are for instance to host a Cloud-EPCIS in every region or country with potential users. A positive side effect of this distribution is the resulting improvement of reliability. With our architecture, the impact of a server failure is naturally limited to a certain area. Also, Cloud-EPCIS from other regions could temporary replace failed servers. However, adding this functionality is subject to future research.

We have further shown in simulations that our architecture efficiently reduces peak loads at data centers and also allows for exploiting scale effects. That is, pooling resources in our architecture enables more efficient resource utilization than providing resources for each value-chain station separately. This has strong implications on provider models for the infrastructures. Namely, our results show that our service is well suited to extend to product portfolio of cloud providers.

Our solution is a continuous technology and forces no immediate switch. We rather provide a concept for accelerating data access when needed, allowing a technology transition that is transparent to the user. We thereby enable high-speed data access to RFID data that integrates with today's RFID solutions. The improvements in access speed can make the data-on-network paradigm for RFID data feasible in more use cases and thereby help using the full potential of RFID technology. With the presented concept we show how cloud providers can access this market and use cloud technologies to provide RFID data at high speed and low cost. In future work we plan to extend our solution and test the performance in PlanetLab.

REFERENCES

1. Abley, J., Lindqvist, K. (2006). Operation of Anycast Services. IETF Request for Comments (RFC 4786). http://www.ietf.org/rfc/rfc4786.txt
2. Androutsellis-Theotokis, S., Spinellis, D. (2004). A Survey of Peer-to-Peer Content Distribution Technologies. ACM Computing Surveys, 36(4):335–371.
3. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M. (2009). Above the Clouds: A Berkeley View of Cloud Computing, Technical Report, http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html
4. Barbir, A., Cain, B., Nair, R., Spatscheck, O. (2003). Known Content Network (CN) Request-Routing Mechanisms. IETF Request for Comments (RFC 3568). http://www.ietf.org/rfc/rfc3568.txt
5. BitTorrent (2009). BitTorrent P2P Protocol. http://bittorrent.org/
6. BRIDGE (2009). BRIDGE WP2 – Serial-Level Lookup Service. http://www.bridge-project.eu/index.php/workpackage2/en/
7. Diekmann, T., Melski, A., and Schumann, M. (2007). Data-on-Network vs. Data-on-Tag: Managing Data in Complex RFID Environments. Proc.of the 40th Annual Hawaii international Conference on System Sciences (HICSS), IEEE Press.
8. EPCglobal (2007a). Pedigree Ratified Standard – Version 1.0, January 2007. http://www. epcglobalinc.org/standards/pedigree/
9. EPCglobal (2007b). The EPCglobal Architecture Framework – Version 1.2, September 2007. http://www.epcglobalinc.org/standards/architecture/
10. EPCglobal (2007c). EPC Information Services (EPCIS) Version 1.01 Specification, September 2007. http://www.epcglobalinc.org/standards/epcis/
11. EPCglobal (2008). EPCglobal Object Naming Service (ONS) 1.0.1, May 2008. http://www.epcglobalinc.org/standards/ons/
12. Fabian B., Günther O. (2007). Distributed ONS and Its Impact on Privacy. Proc. Of IEEE International Conference on Communications.
13. Fabian, B. (2009). Implementing Secure P2P-ONS. Proc. of IEEE International Conference on Communications.
14. Fosstrack (2009). Free and Open Source Software for Track and Trace. http://www.fosstrak.org/
15. Ivantysynova L., Ziekow H. (2008). Design Guidelines for RFID-Based Applications in Manufacturing. Proc. of the European Conference on Information Systems (ECIS).
16. OMNeT (2009). OMNeT++ Simulation Library and Framework. http://www.omnetpp.org/
17. PlanetLab (2009) – An open platform for developing, deploying, and accessing planetary-scale services. http://www.planet-lab.org
18. Saroiu, S., Gummadi, K., Dunn, R., Gribble, S., Levy, H. (2002). An Analysis of Internet Content Delivery Systems. SIGOPS Oper. Syst. Rev., 36 (SI):315–327, 2002.
19. Su, A., Choffnes, D., Kuzmanovic, A., Bustamante, F. (2006). Drafting behind Akamai (Travelocity-based detouring). Proc. of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (ACM SIGCOMM ’06), pp. 435–446, New York.
20. Williams J., Sanchez A., Hofmann P., Lin T., Lipton M., Mantripragada K. (2008). Modeling supply chain network traffic. RFID Technology and Applications. p. 74
21. Zhang, B., Ng, T.S.E., Nandi, A., Riedi, R., Druschel, P., Wang, G. (2006), Measurement-based Analysis, Modeling, and Synthesis of the Internet Delay Space. In: 6th ACM SIGCOMM Conference on Internet Measurement (IMC ’06), ACM Press, pp. 85–98.
22. Ziekow H., Fabian B., Müller C. (2009). High-Speed Access to RFID Data: Meeting Real-Time Requirements in Distributed Value Chains. Proc. of OTM Federated Workshops and Conferences 2009.