TeraCortex

Business and Technology Vision

Subscriber Data Management

February 2018

Abstract

- **The telecommunication industry is faced with new challenges due do changed user behavior and falling margins in traditional business models.**

- **Future subscriber data management systems will see thousand times the usual load.**

- **The current generation of subscriber data management systems is not prepared to catch up.**

- **The TeraCortex database DVTDS is designed to serve these requirements. It is the next generation.**

- **Our solution provides the most cost effective way to scale to next generation subscriber data management.**

Strategic Vision

Welcome to our white paper about subscriber data management in mobile networks and database technology. As a database vendor we look at mobile network evolution from a specific angle.

- **How do changes in behavior and expectations of mobile users influence the requirements towards a subscriber data management (SDM) system?**
- **What specific features are needed to fulfill these requirements?**
- **How does this affect mobile operator core network technology?**
- **How does this affect user data repositories?**

The sequence of these questions is not arbitrary. The pivoting points are the mobile users and their needs, then the operators serving them. Our job is fulfillment at database level. In this white paper we will analyze these questions and give answers.

In recent years a couple of trends emerged that have drastic impact on how SDM systems should handle actions of subscribers. Some of these trends are about change of business models, others are about quantities of traffic. All together they lead to a thousand fold load against databases. This means massive gear up of hardware and sophistication of technology. The current generation of databases is not at all prepared to deliver this in a cost effective way. In the following chapters we will first examine these trends in more detail. Later on we derive a set of requirements a next generation database product must match to make a SDM system future proof. In a product overview we show why our database product is the best and most cost effective choice to meet the future functional and performance requirements. Readers who want to know, how this works out on a detailed technical level should refer to our white paper about architecture and features of our product. You find the sources in the references section of this document.

Now, in a nutshell let’s first summarize the most import trends in the mobile telecommunication industry:

- **Business value shifts** away from the point where mobile devices check into the network and are connected to somewhere. This is done. Simple telephony is done along with the network and database equipment to solve this task. We call this the *access* layer. It is needed to attach to the network. But the today’s party is going on in social networks, Skype, Whatsapp, online gaming, twitter and others. We call this the *service* layer where customers use higher level applications. Here *data* becomes the predominant resource to deal with. This is the point where business value is moving to.

- **Operators need to cut costs** because on the revenue side they failed to develop such attractive high level applications and related business models. Further they are faced with harsh competition in their traditional telephony and SMS business. One means to reduce expenses is the virtualization of mobile network components. Instead of running them on specialized hardware they are moved to virtual machines in the cloud. Their administration can be centralized. Reservation of hardware resources for the worst case peak load is not needed anymore because in the cloud resources are shifted on demand to the point where they are needed. This is the main business driver for virtualization of mobile network functions.

- **Users don’t want to get disconnected** in the middle of a data session. You can easily resume to a broken phone call. A broken bank or stock exchange transaction, a corrupted download or even loss of rank in online gaming is not accepted. Luckily this can be ensured as a byproduct of mobile network function virtualization (NFV). Failed components can easily be replaced when they run in virtual environments instead of using bare metal hardware. Ideally this happens so fast, that the user is not aware of it.

- **Real time** data communications like stock exchange transactions and online gaming users expect very short response times. Complex mobile networks with several hops and long distance communication cannot catch up with these requirements, simply because of the limited speed of light. As a consequence network functions and storage are **located geographically near** the users. When they move around, in a car or whatever, they are automatically connected to the network functions and storage system which is responsible for the new location. This is called mobile edge computing (MEC).

- **5G and the internet of things** are not knocking at the door anymore. They are already in. Besides bringing a **huge increase in traffic** and data to be stored they will also double the number of connected devices within the next five years.

- **User behavior insight** and prediction becomes a valuable asset. Large scale data analysis delivers the tools.

It is clear that all this boils down to the management of data and the question what a particular database implementation can do to handle it in an optimized way. Hardware evolves not fast enough to keep to the pace of exponential growth of devices and data. Therefor databases must be distributed across more nodes. Which calls for linear scaling. The illustration below shows the driving forces.

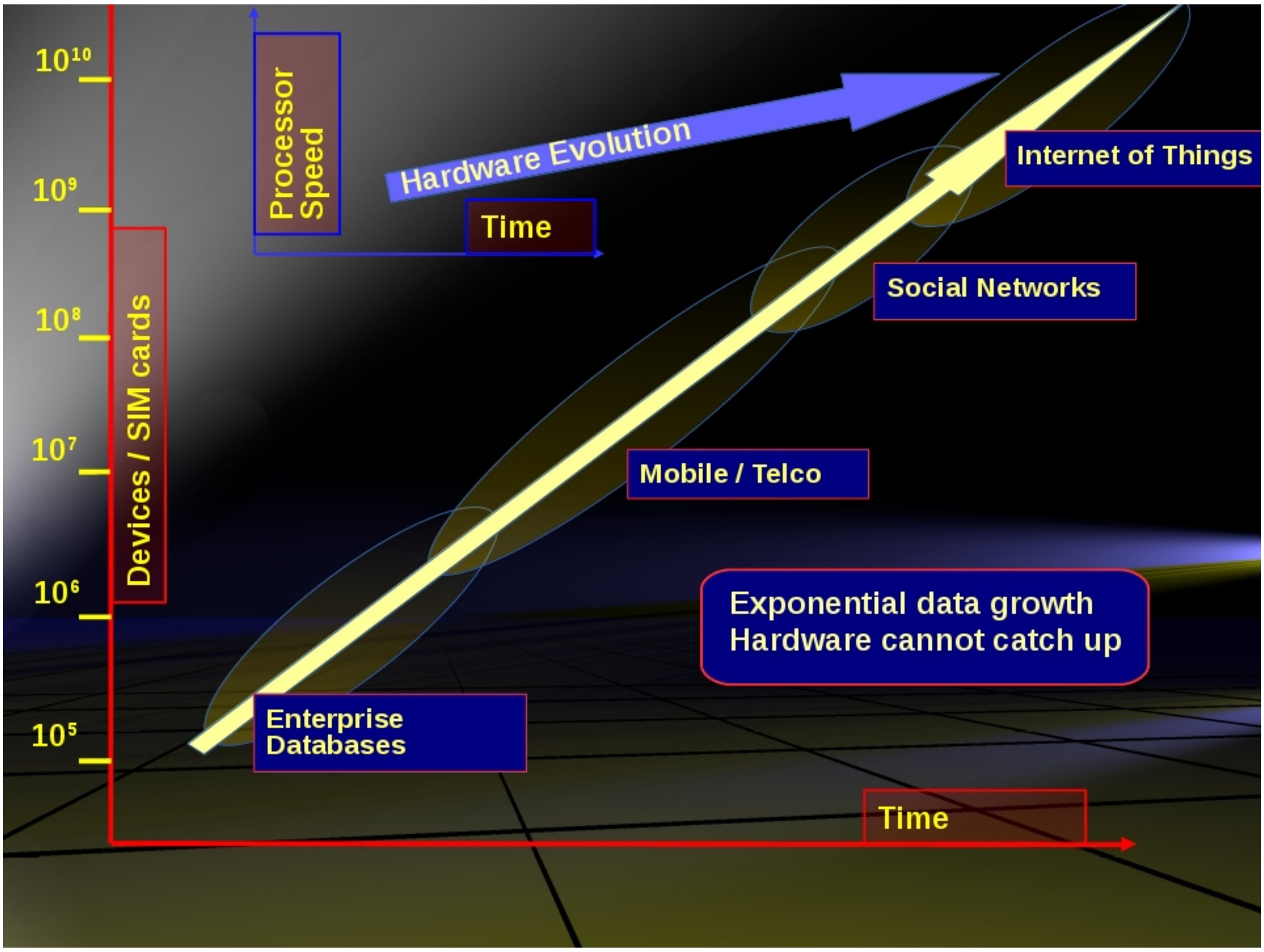

Illustration 1: Evolution of hardware and devices

In the following we will dive deeper into these topics and derive a set of requirements the data base core of a **next generation SDM** system must fulfill.

Shift of business value

Operators and equipment suppliers stick to low margin business fields

For quite a long time **mobile operators** and their equipment suppliers got **good revenue** for doing a good job in providing **voice** and **SMS** service. When operators started to provide access to **data services** things became more difficult. Since then subscribers used *access* ever more excessively to enjoy *services* that are not under control of the operators and do not contribute to their profit: Skype, Facebook, Whatsapp, E-Mail, gaming. This shift of user behavior together with harsh competition in the traditional telephony service faced operators with ever smaller margins in their bread and butter business. Why did former ***startup*** **companies** like Facebook or Whatsapp develop these new hype applications? How could it happen that international **multi billion dollar operators missed the train?** Lack of knowledge? Lack of intelligence? Lack of eagerness?

Not at all. The problem is company culture and the way how operators were used to structure their business over decades. They i**nvested billions** in their frequency licenses, antennas and networks and offered a service that *forced* their customers to use exactly that **equipment**: Voice and SMS. When margins began to decrease they invented tariff options like friends and family and other things. This was nice and might have worked out in some cases. But it represented just a variation of the same attitude towards the customer: We bought equipment. How can we urge you to contribute to amortization? This mind set does not think business from the **user *needs* perspective.** It perceives it from the ***our assets*** point of view.

Then the game changer appeared: The **smart phone**. And suddenly operators were not perceived anymore as most important service provides. Their role was cut down to that of a simple gateway into the wonderful multi media and social network world. Skype and Whatsapp entered the scene and operators tried to fight against with things like the multi media capable rich communication suite (RCS). But as before, this was a service that forced customers into the operator’s equipment and business model. Same mistake: The design was thought from angle of operators *assets* and **not from customers *needs***.

Today the big business is in the social networks and operators find themselves trapped outside a loud and sexy party they are not invited to. They deliver **internet *access*** but ***no attractive service*** on top of that.

The operators are one side of the coin. On the other side sit their equipment suppliers, among them a handful of the **largest companies** which serve over **80%** of the worldwide SDM market. They have a corresponding problem: **Falling margins** in the SDM business. A company culture centered around delivery of mobile core network components: Antennas, transceiver stations, switching centers, network gear and business logic applications. An attitude that perceives their central subscriber repository database just as a tool to solve a telephony problem. Consequently they designed their **databases as an appendix** of mobile network applications like the home location register (HLR). But they did **not design it as a general purpose** product for business areas *beyond* the access layer of mobile networks. Like a Facebook subscriber DB. A Twitter subscriber DB. An ultra fast real time DB for online gaming. They *use* a database to fulfill the particular task of connecting mobile devices. But they ***are no* database companies.** And same as the operators they behave as being trapped outside the social network party. Naturally two sides of a coin belong to the same piece of metal.

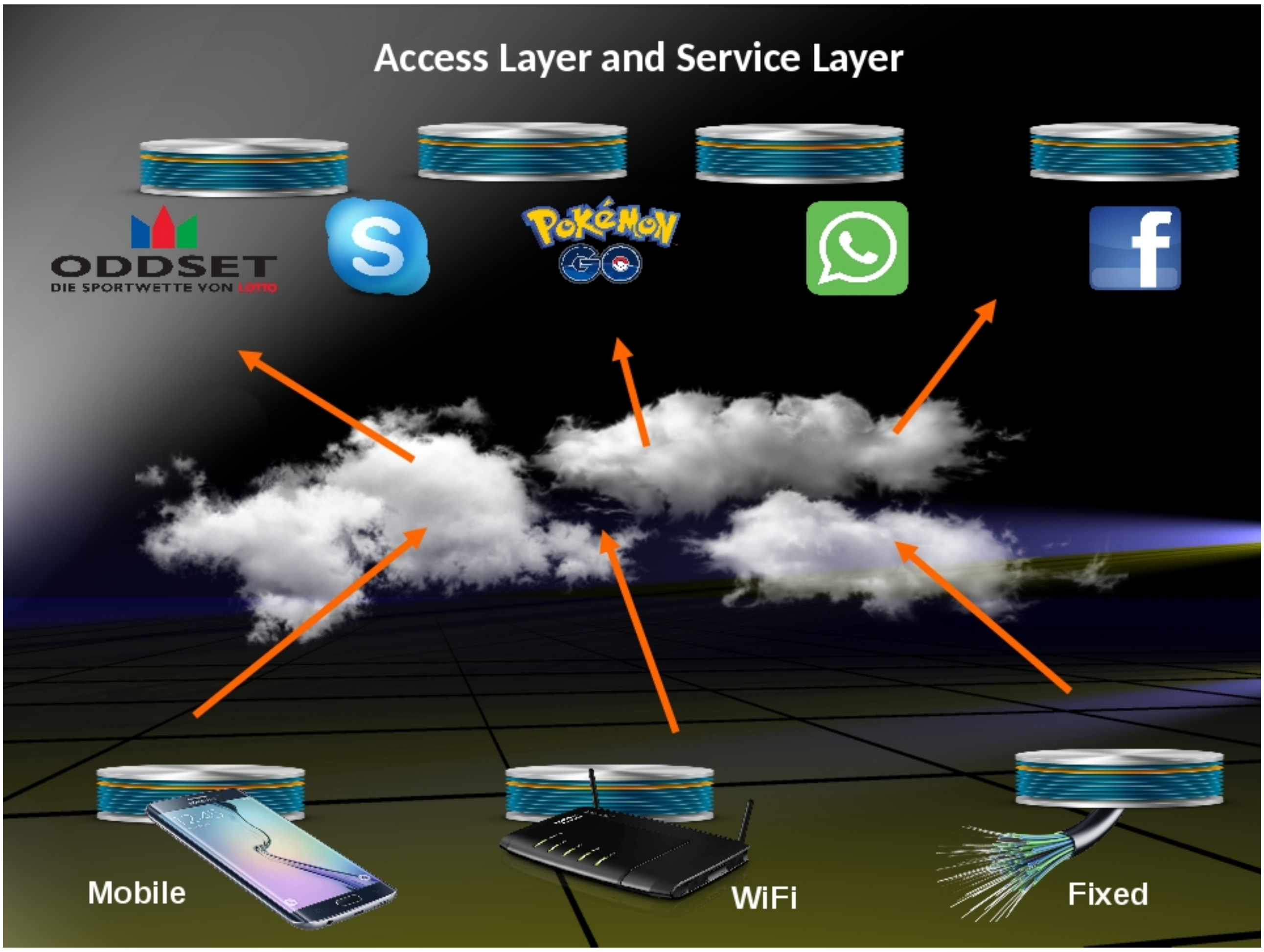

Illustration 2: Access layer and service layer

The illustration above gives an idea about the relation between access layer and service layer. The lower part shows the access layer, where operators provide a core network and SDM databases to handle authentication and subscriber profiles. The upper part depicts some of the things smart phone users are interested in. The picture already tells a first important part of the story. We get the second part when we combine it with some insights about user behavior, found in the November 2017 edition of the Ericsson mobility report.

In the picture below we see almost constant (and by amount almost irrelevant) voice traffic versus exponential growth of data traffic between 2012 and 2017. In the upper left the compound annual growth rates for the most important categories of data are shown, projected until 2023. Here video on demand and social networking are believed to grow fastest. ***This*** is the market where the **music plays.**

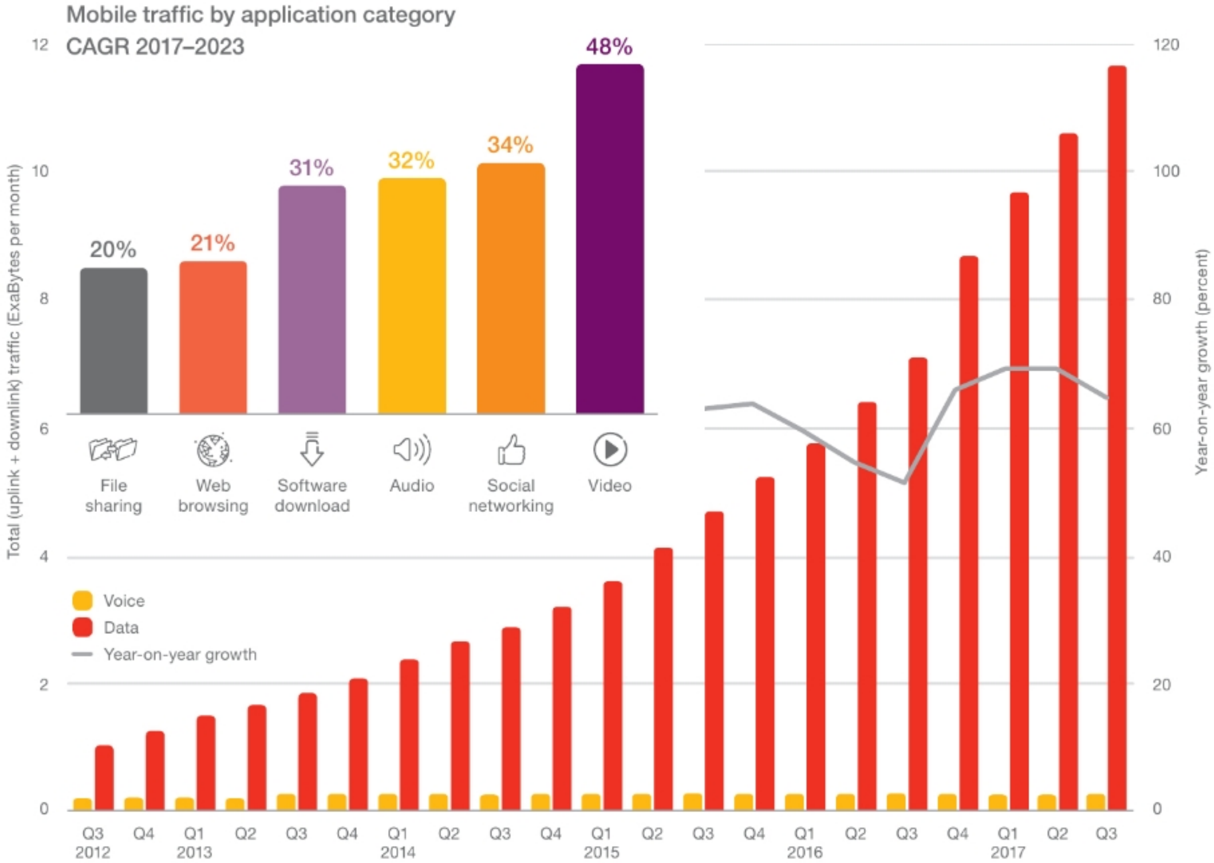

Illustration 3: Voice and data traffic

How could you, the equipment supplier participate?

- **Invest in database technology able to handle the future SDM load**

- **Take care for large scale data analysis**

- **Make operators aware of benefits of next generation SDM architecture**

- **Look beyond your traditional SDM business: Logistic and social networks, scientific environments, industrial automation, real time event capturing in industrial processes and security systems.**

Operator Cost Savings

Operators cut costs due to smaller margins in traditional business fields

Telecommunication providers increasingly **moved away from specialized hardware** formerly delivery by their equipment suppliers and used to run their mobile core networks. Such hardware is more and more replaced by cheaper standard rack servers called commodity of the shelf (COTS). This process is still going on since a couple of years, but gets now super seeded by the next round in cost reductions. Core network application software is removed from bare metal computers. Instead the functionality is implemented running in **cloud based** virtual machines or containers. Further the network element specific data silos **are removed** in favor of a central database that provides **shared data** for all kinds of core network elements. Besides purchase **price considerations** this approach yields a set of benefits:

- **Easy scaling.** This enables a flexible response to **load peaks**. Resources can be shifted dynamically inside the data centers to cover sudden peaks of daytime, situation or application specific traffic. This leads to a smoother and more averaged use of equipment and lowers the overall amount of needed hardware.

- **Centralization of network component** and functions management. Operational data produced by network components is not stored or monitored anymore locally or component – wise. Instead it is transferred into a central database where it is available for generalized monitoring and data analysis.

- **Network functions** dealing with online session data of mobile users need **no local storage anymore**. Instead they outsource their internal states and sessions to a central database. When such a virtual network component fails the cloud management immediately brings up a **replacement** of the same type. That one can pick up latest state and user sessions from the central database and continue the service towards the user. Ideally the user notices no interruption of service. To a certain extend this avoids registration storms because in case of disconnection millions of users would frantically try to reconnect. By avoidance of such peak loads the overall dimension of the core network can be lowered which leads to **substantial cost savings.**

- **The communication** between different elements of the core network is drastically **reduced** because elements can pick needed data from the central database instead of sending or requesting it to / from other elements. This **simplifies the architecture** of the core network architecture and communication matrix between its elements. Further it eases maintenance.

From a business perspective these moves are reactions to falling margins in the mobile telephony and SMS business. They intend to **cut costs** in operation of networks and databases because on the revenue side operators are limited by their restrictive traditional business models and saturation of the related markets.

However, these cost savings do not come for free. They pay off only under the assumption that the central database is powerful enough and cost effective in terms of hardware utilization and administration effort. Especially the operational data and the user session data from virtual network components produce a very high workload.

Example for the technical interested reader:

Imagine a component counting down used data volumes for millions of concurrent users while they are using their devices. This core network function is needed for billing and is called the policy charging relay function (PCRF). Future PCRF will keep this information in the central database at very high request rates. Let's assume just one million people (which is only three percent of customers of a medium sized operator) having mobile internet connections at the same time. Let's assume the PCRF registers the consumed data volume every second for every connected subscriber. Then this means one **million database transactions per second.**

Low Response Times for Applications

Mobile edge computing yields lowest response times

Mobile applications require ever **shorter response times.** It becomes more and more difficult to serve these needs because the transfer of requests through the core network and from there to central user data repositories (UDR) is limited by the **speed of light**.

The Solution: Instead of bringing the request to the UDR, the UDR is **brought near the user.** More precise: The network functions and partial data needed to fulfill the current task (usually the user profile along with user session data) is brought **geographically near** the user. Just imagine the mobile core network as a big bubble where some equipment does useful things. And imagine users attaching the bubble somewhere. These interfaces are called the **mobile edge.** Mobile edge computing **locates** parts of the core network equipment and storage in **close proximity** to these interfaces. It depends on operators how far they want to drive the concept. In the extreme case a small UDR along with needed core network elements could run in small boxes in the base transceiver stations (BTS) that control cellular antennas. But operators may also decide to setup a bunch of local data centers serving defined regions.

The purpose is the same: **Response times go down,** when users can attach to the needed equipment more locally and the signals have only a **short distance** to travel.

The above concept assumes a two tier approach with one database layer at the mobile edge and one central UDR. However, for larger networks this can be turned into a hierarchical deployment with local, regional, national and global database layers. The global layers would implement the handover / roaming to networks of other operators. This ensures that even subscribers roaming in foreign networks enjoy the short response times of mobile edge computing.

Consequences for a Database System

The current generation of database lacks the needed functionality

The **new trends** and business cases have a large impact on the required functionality and performance of a **next generation database**. In the following we outline this and give more background information along with some reasoning. With the exception of some sub features in the first point the requirements are not supported by the current generation of SDM systems. Necessarily the following discussion goes into deeper **technical detail.**

1. The **classic subscriber data management** at the access level must of course be supported. This includes the handling of subscriber profile data along with transaction safety, geographic redundancy and reliability, combined with real time capability to serve core network elements.

2. Many different **core network applications** use these repositories in **shared mode:** They have their own *logical* view on the data which is stored (aside from replication) only once in a certain *physical* layout. For this reason the database must provide application – specific data views that translate the physical data format and structure to the logical model the particular application wants to see. Further the 3GPP standard compliant access protocols (LDAP, SOAP) must be supported.

3. **Multimedia data dominates** the storage requirements. Due to the large price gap between RAM and hard disk based storage this data cannot be kept in main memory at reasonable cost. But it must be available for any time access. The database must keep it in fast caches when needed and must offload it to disk when it is currently not used. This means that certain *categories* of data must be stored in a specific way. The behavior should be transparent to the clients. They need not care how specific data is stored. And they should see it seamlessly integrated in their data model regardless of the underlying storage technology.

4. **Real time capability is needed** to respond to actions of millions of users. A perfect example is the *Pokemon Go* hype in 2016. In this game players move with their mobiles in a real geographic landscape and catch there virtual goods displayed on their screens. It is rapidly changing individual information where a player moves in the scene, which gadgets are in reach and which ones he or she already got. This means that the database must track their geographic locations at the highest possible resolution in real time and match them against the positions of the virtual gadgets and other players. The used system (Google data store) was not all suited to perform this simultaneously for such an amount of users. It readily broke down under fifty times overload, which ended the hype.

5. Due to the **high number of mobile users** the handling of their session data leads to extremely high request rates, about **one thousand times** compared to today.

6. **Real time capability** usually excludes the use of hard disks. Only the main memory is fast enough to deliver responses within micro seconds. This means again that certain *categories* of data must be stored in a specific way, in this case in RAM. And as with other categories of data like multi media stuff clients should not need to care about the type of storage and see this data seamless integrated within their logical model.

7. **High level service layer applications** like the ones described do not care for mobile network standards like 3GPP nor LDAP nor SOAP. They use more recent protocols like REST, BSON or CQL. In addition the 5G specifications foresee JSON over HTTP as additional interface. The database must offer appropriate interfaces. In most cases database actions boil down to creation and deletion of tables and indexes, authentication and creation, read, update and deletion (CRUD) of data. The specific interface must map these operations to the internal storage engines and structures. The goal is to keep an internal consistent data image and structure regardless of the particular access protocol.

8. Given the sheer amount of subscribers the system must **scale storage capacity and throughput in linear fashion.** This excludes the old X.500 style hierarchical arrangement of server nodes where a central component stores all keys and a set of subordinate components store the data. Instead it requires the use of equally distributed indexes and data. The distributed hash table (DHT) architecture first proposed by Amazon is the right choice. This architecture also supports **easy scale** in and out **without service interruption.**

9. Same as with storage the **replication of data depends** also on its ***category*.** Persistent data like subscriber profiles is not changed frequently. It can be copied to geographic redundant sites using the slower but reliable synchronous replication. Due to network latency this is not possible with high frequency data like the one generated in real time online gaming or user session data of mobile core components. Such data should be replicated in asynchronous mode or only within the same geographic site.

10. Replication of high volume multi media data requires very high network bandwidth. In case where geographic redundancy is not needed it is sufficient to replicate to remote sites only a link to such data, instead of transferring the full volume. The database should support such **special handling data category – wise.**

11. **Data analysis** provides **insight** into user behavior and market trends. This enables operators to implement additional value chains. The database must provide high speed parallel interfaces to data analysis sub systems.

Beside these fundamental properties a database system must feature many **more smaller or bigger functionalities**. In the following chapters we describe how our product solves the given tasks. With small exceptions it fulfills all of the above requirements, giving it a head start in the race between next generation database systems.

Moreover, it has one distinct advantage: It is about **ten times faster** than the competition. When talking about **many million requests per second**, it makes an eight digit difference whether you buy **thousand rack servers or just hundred**. This gives our SDM partners a **huge margin** in negotiations with telecommunication operators.

The table on the next page summarizes how our database supports the next generation use cases and business models. The green fields are implemented and tested. The yellow ones are under development. Please note, that this overview presents only the high level features. Readers who want to compare our database on detailed product feature level should refer to our DVTDS description document. You find the references at the end of this document.

Afterwards goes the discussion deeper into technical details of benchmarks and improvements of throughput and response times. Readers not interested in this level of detail may skip to the conclusion at the end of this document.

Trend	User benefit	Operator benefit	Supplier benefit	Database feature
High level services	Social networks	Increase revenue	New customers Growing markets	Scaling data
				Multi media capable
				Multiple protocols
	Online gaming			Microsecond response
	Multi media			Any data type
	Real time apps			Scaling throughput
Network Function virtualization	No interrupts	Cost reduction	Sell equipment	Data category specific storage
		Avoid signal storms.		Data category specific replication
		Better resource utilization		Automatic scaling
		Central administration		Cloud ready
Mobile Edge Computing	Shorter response times	Attract customers. Increase revenue.	Sell equipment	Special replication capability
Data sharing		Multiple applications, Single repository		Data views
Data Analysis	Tailored services	Insight to user behavior and market trends.	Sell equipment	Data analysis interface
Central SDM		Secure data handling	Replace equipment	Traffic encryption
				Access control
				Distributed transactions

Table 1: Trends, benefits and features

25 Million Requests per Second

TeraCortex DVTDS delivers world record performance

In 2015 we conducted extensive benchmarks of our database in the Amazon cloud. Of course we fulfilled the functional requirements. Our product delivers in much less than a millisecond. Simple requests with not too much data are served in less than 100 microseconds. We distributed more than four billion keys across a set of just 16 virtual machines and yielded around 22 million transactions per second. Mixed read write access (non transaction) topped out at 25 million per second, both with the option of further linear scaling. See the picture below.

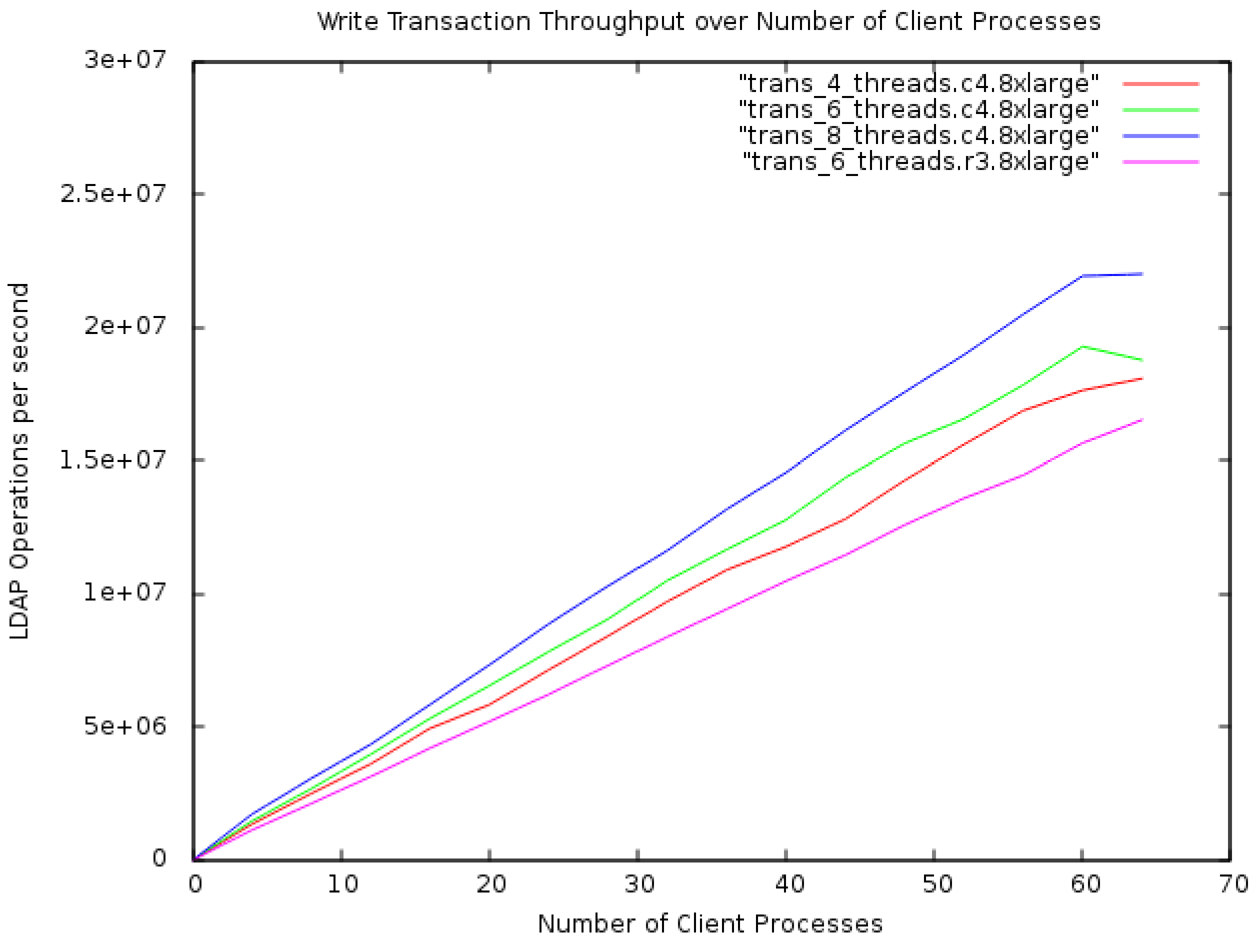

Illustration 4: Cluster benchmark results

Increasing the Throughput

Vectorization triples the throughput and yields one third of average response times.

It is an old insight that throughput scales, when a single instruction is applied in parallel to multiple data. This is known as SIMD (Single Instruction, Multiple Data) and on basic hardware level every computer does this million times per second. The principle can be transferred to SDM systems when looking at the data model of UDR. A mobile subscriber has a defined structure and the storage keeps millions of them. The point is: They all look similar with small variations according to their booked services.

Given this data model layout a core network element could collect its inputs originating from actions of different users. It then applies the defined business logic to *either* of them, creates the associated database operations for *either* of them and then send the entire set of operations in a *single* action to the database. The database applies the operations to the data records of all affected users and returns the responses to the client.

One element of the vector is given by operations against the data record of a particular subscriber. This subscriber is identified by a key, IMSI, MSISDN, whatever. Other elements of the vector belong to other subscribers. The great advantage of this concept is network efficiency. A large vector of operations can be compressed with a much better ratio than individual operations. Further the network latency appears only once on the bill for the entire set instead of showing up for every single operation.

What works out for the immediate client of the database can be a concept for the entire communication chain across several core network elements. One could object that processing an entire vector takes much longer than working on a single request, thus increasing response times. This is true for the immediate client of the database, for the client of the client and so on. But at the mobile edge towards the user devices this bigger response time splits into as many pieces as their were users in the vector operation. The sum of these pieces is much lower than the summarized latency of as many individual operations.

To give an impression: A fast and simple update operation in our database takes about twenty microseconds (which is already beyond good or evil). The throughput is in the order of 47000 operations per CPU core. When using vector mode the throughput triples to 140000 per second and CPU core. The response time per operation falls to seven microseconds.

The picture below shows an example of a benchmark we conducted in 2015. It proved how powerful the described techniques are when fully implemented. Three databases replicating to each other in multi master mode were deployed **across Europe and the Americas.** At either site parallel clients flooded the local nodes with updates at maximum speed. Either of them replicated to the two other sites. Using vector mode we achieved more than **one million updates per second** in *synchronous* mode. Across the Atlantic ocean using lines with more than 200 milliseconds latency. We tried hard to find in the internet any other database benchmark report at this scale. No avail.

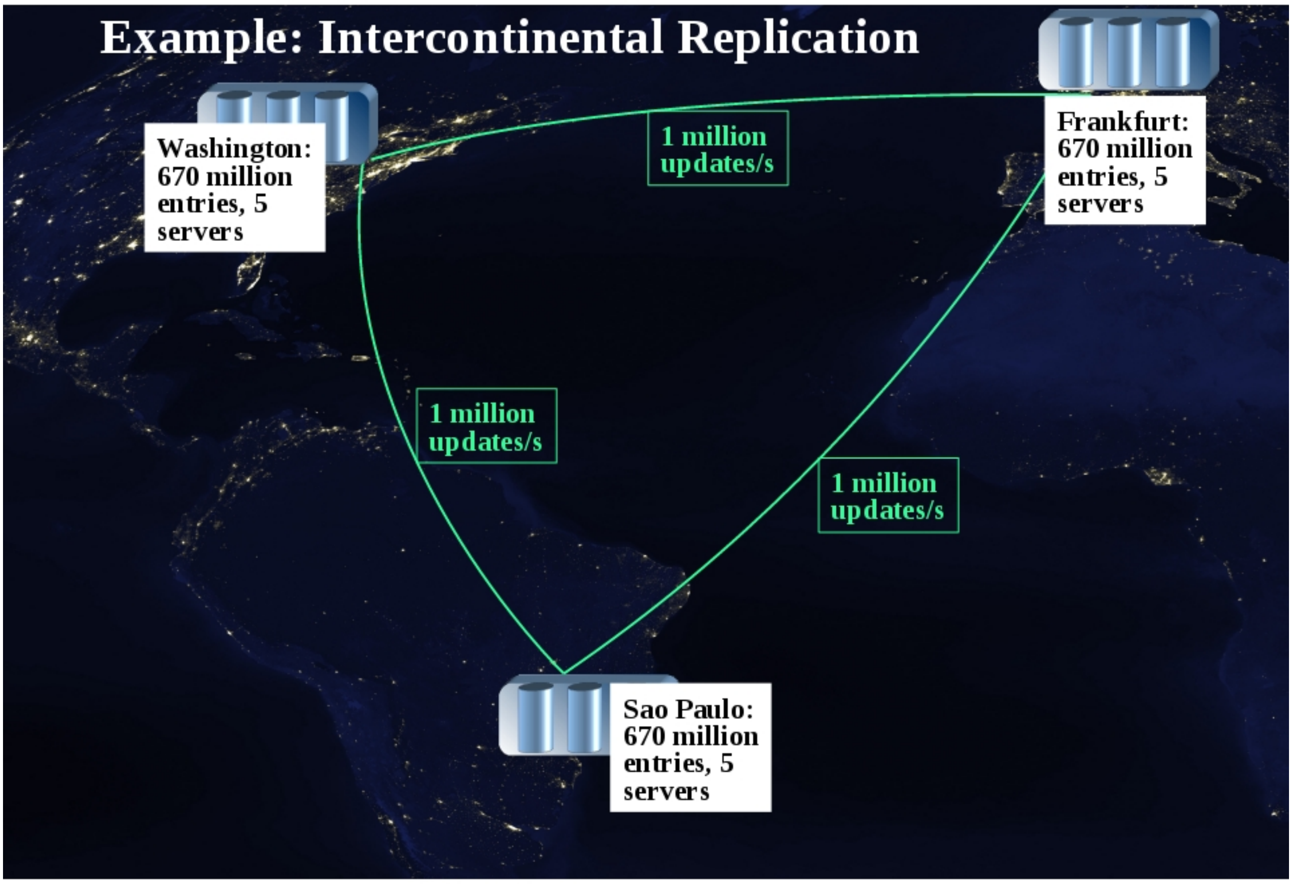

Illustration 5: Intercontinental replication

For the database this means the following:

1. It supports vectors of operations against arbitrary collections of keys and their associated data.

2. The execution of such vectored operations does not conflict with internal queuing mechanisms or particular implementation of transactions. Nor does the database make any assumptions about the structure of the data model and how an “appropriate” operation should look like to match that structure.

3. For network efficiency the collection of responses to a vector operation is packed into a single buffer.

4. Replication uses the same vectoring mechanism, means: Updates contained in the vector are applied in vector mode to the replica. If parts of the data are subject to special consistency policies (synchronous / asynchronous) the vector is split into subsets and either subset complies with these policies.

5. Given the size of SDM systems the data is normally partitioned across a set of nodes. Particular operations in the vector are likely to target different nodes. This requires also appropriate split of the vector into several subsets. When receiving the responses from the peer nodes the originating node merges them to a single set of responses towards the client.

Mobile Edge Computing

The data walks with the user

As shown earlier in this document, mobile edge computing is a means to ensure **shortest response times** in mobile networks. In terms of replication and reaction to client requests this requires special support and behavior in the database. As with most requirements discussed in this document mobile edge computing is **not supported by the current generation of SDM systems.** The points below summarize the way how we implement this.

1. When a client access comes with a key that is not found, the mobile edge database does not simply reject the request (as is today in established SDM systems). Instead it retrieves the associated profile from the central UDR and makes it available for client access in the local storage. Depending on the amount of data this may take a short while but in any case it takes only marginally longer than letting the client retrieve the same data directly from the central UDR.

2. The mobile edge database replicates all local updates to the central UDR in asynchronous mode. The client may override this by using an LDAP control. The control tells to which other nodes updates are replicated and whether this happens in synchronous or asynchronous mode. The control works on a per request basis or is valid for the entire client session, when it was used in the LDAP bind operation. For replication directives our product implements a precedence driven approach. Directives in individual requests precede directives valid for the entire client session. Client session level directives precede statically configured directives associated with categories of data which in turn default replication configuration.

3. When the user changes location to a point that is served by another mobile edge instance, the local client application may use the LDAP control to enforce for the last update synchronous replication to the central UDR. This ensures consistency before the user checks into the next instance of the mobile edge when that instance retrieves the data from the central UDR. However, this is left at the disposal of the local client. Asynchronous replication is normally done in a fraction of a second. When the user arrives in the new cell, the last replicated update in the central UDR should be done. If it is still ongoing, the retrieval of data by the next mobile edge instance is blocked anyway because it waits for the update write lock to be released.

4. When the user attaches to the next mobile edge instance the process is repeated there. In summary the data walks with the user. It is always near to the user location.

5. The concept gives superior response times as long as a user moves only within an instance of the mobile edge. It cannot avoid the usual latency when a user enters or leaves an instance. However, these effects can be mitigated when high data volume parts of the needed contexts are stored in the mobile device while it it moving. The mobile device can carry them instead of repeatedly transferring them over the network.

Conclusion

The discussion shows that telecommunication industry and the SDM suppliers serving it have come to an inflection point. **Diminishing margins** in providing network access on one hand. On the other hand **emerging technologies** like network function virtualization with performance requirements that were beyond imagination just a couple of years ago. On a third coordinate the need for complex database functionality depending on the category of data. Existing designs and products are constrained by architecture, implementation, performance or licensing issues, making it difficult to shape them to the format required here. This calls for a **next generation SDM database** able to do the job.

From its very beginning our database was designed to make maximum use of parallel resources. With decades of telecommunication experience in our bones we refined it carefully, tuned its performance and added innovative functionality whenever the future popped up in form of new requirements. **Today our database fulfills almost all of the discussed requirements** and offers a rich set of features beyond them. Additional functionality is under development.

Our database DVTDS was developed from scratch by TeraCortex with the final objective to make it the **most powerful SDM server** on the world market. DVTDS has a rich feature set missing in other solutions. It runs stable with the current functionality and it outperforms state of the art LDAP servers like Isode MVault, OpenLDAP or Oracle OID several times. Benchmarks have also shown superior performance over well established NoSQL databases like Cassandra, FoundationDB and others. Currently there is **no comparable product on the world market.**

You find more details and features in our technical data base description. For a deep dive into performance please refer to our benchmark documents listed in the references below or take the first link when googling for ***LDAP world record*****.**

References

No	Document	Description	Where
1	Features	DVTDS technical data description	Confidential. Available on request
2	Cluster Benchmark	DVTDS cluster performance	www.dvtds.com/en/download.html
3	Global Replication Benchmark	DVTDS replication across five continents	www.dvtds.com/en/download.html
4	50 Million Entry Benchmark	DVTDS single node benchmark	www.dvtds.com/en/download.html
5	Slide deck	Presentation at international LDAP conference 2013 in Paris, France	de.slideshare.net/ldapcon/distributed-virtual-transaction-directory-server
6	Slide deck	Presentation at international LDAP conference 2015 in Edinburgh, UK	ldapcon.org/2015/accepted-papers/improvements-of-ldap-protocol-and-transaction-protocol/

Table 2: References

Glossary

Abbreviation	Meaning
3GPP	3th Generation Partnership Project
ACID	Atomic, Consistency, Isolation, Durability. Transaction paradigm
API	Application Programming Interface
BLOB	Binary large object
BSON	Binary jSON, MongoDB protocol
BTS	Base Transceiver Station
COTS	Commodity Of The Shelf
CQL	Cassandra Query Language
CRUD	Create, Read, Update, Delete. Standard database operations
CSS	Customer Self Service
DB	Database
DHT	Distributed Hash Table
DVTDS	Distributed Virtual Transaction Directory Server
GPS	Global Positioning System
HLR	Home Location Register
HTTP	Hyper Text Transport Protocol
IMS	IP Multimedia System
JSON	Java Script Object Notation
LDAP	Lightweight Directory Access Protocol
OAM	Operations Administration Maintenance
PCRF	Policy Charging Relay Function
RAM	Random Access Memory
RCS	Rich Communication Suite
REST	REpresentational State Transfer
SDM	Subscriber Data Management
SIMD	Single Instruction Multiple Data
SMS	Short Message System
SOAP	Standard Office Access Protocol
SQL	Structured Query Language
SSD	Solid State Disk
TLS	Transport Layer Security
TTL	Time To Live
UDR	User Data Repository
X.500	Protocol and Directory standard

Table 3: Glossary