must make information easily accessible;
must present information consistently;
must be adaptive and resilient to change;
must present information in a timely way;
must be a secure bastion that protects the information assets;
must serve as the foundation for improved decision making.

Different data warehousing systems have different structures. Some may have an ODS (operational data store), while some may have multiple data marts. In view of this, it is far more reasonable to present the different layers of a data warehouse architecture. In Figure 1.1, a data warehouse layers are depicted.

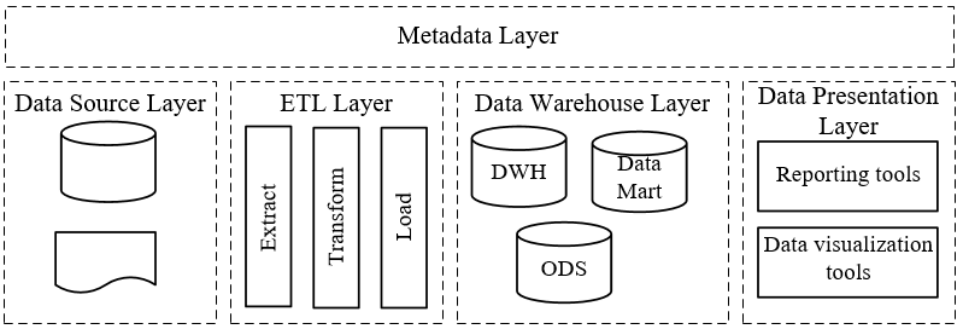

Figure 1.1 – Data warehouse layers

A data warehouse architecture exhibits various layers of data in which data from one layer is derived from data of the lower layer. The lowest layer is a Data Source Layer. This represents the different data sources that feed data into the data warehouse. The data source can be of any format -- plain text file, relational database, other types of database, Excel file, etc. The second layer is an ETL (Extract-Transform-Load) Layer. Data is pulled from the data source into a Data Warehouse Layer by means of ETL process. There are three components in the data warehouse layer, namely operational data store, data warehouse, and data marts. Data flows from operational data store to data warehouse and subsequently to data mart. A Metadata Layer stores information (metadata) that refers to data about data. It describes where data are being used and stored, the source of data, what changes have been made to the data, and how one piece of data relates to other information. The highest layer is an End User Layer consists of tools that display information in different formats to different users. This can be in a form of a tabular / graphical report in a browser, an emailed report that gets automatically generated and sent everyday, or an alert that warns users of exceptions, among others.

Large organizations today need flexible access to various kind of information that is present in its operational systems. The data warehousing technology facilitates creation of integrated and subject-wise history data, and provides flexible ways to access, aggregate and visualize the information. This paper introduced basic layers for building a data warehouse.

References:

1. Kimball, R. The Data Warehouse Toolkit: The Defi nitive Guide to Dimensional Modeling, Third Edition / R. Kimball, M. Ross // Wiley Publishing, Inc. – Indianapolis, Indiana, 2013 – 564.

2. Inmon, W. H. Building the Data Warehouse, Fourth Edition / W. H. Inmon // Wiley Publishing, Inc. – Indianapolis, Indiana, 2005. – 576 c.

EFFECTIVE MICROSERVICES

Liubanets A.Y.

Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus

For several years, monolithic architecture has been the widely-used architecture for building web and mobile applications. These applications were mostly characterized by individual programs handling multiple functionalities. Though monolithic applications were known to be easier to operate, as the systems grew bigger, they introduced complexity for both the coding and deployment stages of software development life cycle. Single point of failure, technology lock-in, and limited scalability are a few other drawbacks of monolithic applications. Recently past, microservices architecture has evolved as a paradigm shift in decomposing large monolithic applications into smaller manageable pieces with their own code base and deployment infrastructure. Considering practical business applications, multiple microservices are required to fulfill one single case of usage, still we can describe microservices as simple and stateless.

Well-designed microservices can make our life much easier. From design to development to rolling out to production, it can help improve all the aspects. While microservices have been around for years, and best practices have been evolving during the time, dealing with microservices at a larger scale has not been easy. Sometimes the complexity may be overwhelming for teams for whom microservices are new.

There are nine puzzling pieces of an effective microservices architecture:

1. Containerization

It is almost impossible to build microservices without containerization. There are just too many pieces. Imagine if you wanted to buy some Oreo's and they came separately. It would be less convenient.

When it comes to services, containers simplify development, testing, deployments, and running in production.

2. A cluster

It is very convenient to use more than one server for microservices system. It is like doing the shopping when you put heavy things on the bottom, so they don't squeeze bread in one bag and cold things in another one. So for an effective system each service should be grouped with another in order to make it easier to control them. AWS cloud provider is very helpful in this case.

3. An orchestrator

Let's continue with an example about shopping. In this way an orchestrator is like the grocery bagger which knows the memory requirements of system services, and how many CPU need to be allocated, and it places each service into the appropriate server carefully. Docker Swarm, Kubernetes are known tools in this sphere.

4. Continuous deployment

If a goal product is a fast growing system, continuous deployment will be a time saver for it. The whole Lean Startup is about building, measuring, learning and repeating this cycle. Continuous Deployment allows to do this faster. Then you can measure and learn and know what to build next. Jenkins (with pipeline and Blue Ocean) will be perfect for this.

5. Proxy

Systems services should be secure. Instead, use a public facing reverse proxy. However, it must be easily configurable so the team developing the service can decide much better how to interface with the proxy without having to sash into servers. A proxy often uses a proxy server because it is easy to configure.

6. Message Queue

Services should communicate in a universal language. A Message Queue can be used as the glue that sends messages to all subscribed clients. Also, it should not be a service responsibility to know where other services are running on the internet. Use one-way «fire and forget» messages to make services loosely coupled, and easy to scale. Using AWSMQ, RabbitMQ and so on the system will have such features as retries and error queues, a support team can be stressed less about services crashing. The message will simply be retried when the service comes.

7. Centralized logging

With messages happening in many nodes and many replicas, which are potentially being moved around by the orchestrator, it's not a viable option to sash into a server and check the logs. System needs to ship them to a centralized location where they can easily be searched and driven insight. Furthermore, it is not the service responsibility to know how to ship logs. Instead of it, use a tool to collect all of those logs by running it globally on each node in systems cluster. The log collector tools will then ship the logs to system Centralized Logging subsystem.

8. Monitoring and Alerting

Similarly, to logging, system doesn't want to be chasing services around to see what their usage statistics are. Instead, they should all be collected and shipped to a centralized location. Use alerting to drive scaling or recovery events before alerting humans is very good practice.

Ideally, system should only ever hear a «peep» when something is wrong. This means that the system needs an alerting tool which can respond to the metrics collected from monitoring and alert a user. It can alert a microservice to trigger a scaling event, or attempt to resolve the issue in other ways to avoid requiring human interaction. Prometheus, Alertmanager, Grafana are tools for these tasks.

9. A Microservices Mindset

It is not done when the infrastructure is build. For the whole team to archive microservices's benefits, every member should understand the principles of microservices architecture, so they can approach building with confidence, and less experimentation. The conclusion of the present work is that microservices are not an easy design pattern. It requires a lot of knowledge not to bump into a trap because microservices architecture sometimes is much harder than monolith.

References:

1. Patrick Lee Scott, “Pieces of an Effective Microservice Architecture” https://hackernoon.com/what-makes-a-microservice-architecture-14c05ad24554.

2. Chris Richardson, “Microservices from design to deployment” NGINX

DIGITAL DEVICES ACCESSIBILITY FOR DISABLED PEOPLE

Karmaz A.M.

Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus

Nowadays people with disabilities meet barriers of all types. However, technology is helping to lower many of these barriers. By using computing technology for tasks such as reading and writing documents, communicating with others, and searching for information on the Internet, students and employees with disabilities are capable of handling a wider range of activities independently. Still, people with disabilities face a variety of barriers to computer use.

According to the World Health Organization (WHO), approximately 15% of the world's population lives with some form of disability. Persons with disabilities can equally participate in society and make substantial contributions to the economy if the appropriate Internet tools are available.

Moreover Information and Communication Technologies (ICT) can be a powerful tool in supporting education and inclusion for persons with disabilities. Technological development can enable people with disabilities to improve their quality of life. The successful application of such technologies can allow people to use all existing digital devices.

Over 100 Governments have signed and ratified the UN Convention on the Rights of Persons with Disabilities. Obligations include implementing measures to design, develop, produce and distribute accessible ICT at an early stage, so these become accessible at minimum cost for persons with disabilities.

W3C’s Web Content Accessibility Guidelines (WCAG) are increasingly mandated by governments and used by industry to make websites more accessible for people with disabilities.

More governments are starting to incorporate accessibility criteria in their public procurement policies.

Popular companies now have progressive attitudes to accessibility. Firstly, the Governments incorporates accessibility criteria in its public procurement policy (through what are called Section 508 guidelines) thus stimulating industry to supply more accessible products to its agencies. Secondly, litigation under discrimination and telecommunications legislation has focused companies’ attention on the possible consequences if accessibility needs are not appropriately addressed.

There are new potential technical solutions that can benefit both persons with disabilities and the general community. Products like speech recognition (originally designed for people with limited hand movements) and the scanner (designed as part of a document reading device coupled with speech synthesis for blind people) are now mass market products.

There are internationally recognized accessibility guidelines developed by W3C on web content, authoring tools and user agents. These guidelines, especially the Web Content Accessibility Guidelines (WCAG), are used by many governments to build accessible websites. Version 2 of WCAG stipulates that websites are to ‘perceivable’, ‘operable’, ‘understandable’ and ‘robust’. The guidelines detail how this is done under three levels of success criteria.

Cloud computing has massive potential to deliver affordable and accessible services to persons with disabilities.

Internet technologies have the potential to give persons with disabilities the means to live on a more equitable basis within the global community in a manner that previously was not possible.

The Internet Society’s motto is “The Internet is for Everyone,” reflecting the belief that access to the Internet is a fundamental public policy issue. Apart from access to infrastructure and equipment, accessibility depends on making physical devices and online services useful to everyone, including persons with disabilities.

These are positive developments; but there needs to be ongoing vigilance. Without ongoing efforts to raise awareness, new types of products may create new barriers.

Examples of what is being done across sectors and what the Internet community can do to increase and enhance the use of the Internet by persons with disabilities were offered.

Regardless of the challenges they may face, persons with disabilities can contribute to society like any other member of the community when barriers are removed. Increasing accessibility to the Internet can help to make that happen. Governments, industry and other key stakeholders need to make accessibility a priority in their ongoing work, individually and collaboratively. Internet community should