



Technology Radar

September 2018

About the Radar

Nowadays, in a fast-moving world, it is increasingly difficult to stay up-to-date with technology. From our 'IT' perspective it is important to know what's available on the market - sometimes out-of-the-box and often for free - to not reinvent the wheel or repeat mistakes of others. As Tim Lister, an American software engineer and author, said "You've got to surf the technology wave, or you'll drown". Lastly, for your own career, it is worth knowing in which direction the world is evolving to develop your own skills to gain future work benefits.

We really liked the idea of the technology radar. The radar itself is a circle with dots representing different technologies. Each dot is placed in one of four quadrants logically grouping elements (.NET & Azure, Web & Mobile, Data & AI,

Java & AWS) and in one of four rings indicating its trend (Adopt, Assess, Trial, and Hold).

- The hold ring means to stay away from a particular technology because it has known flaws and/or is not aligned with direction of our projects, teams, etc.
- The assess ring indicates new promising technology, but one that is not fully understood yet, so some investment (like a development spike or reading a book or watching a presentation) is required to assess its usefulness and impact.
- The trial ring is for technology you understand but have never tried in real project (enterprise, in our case), so it should be proven first in a Proof-Of-Concept or small project

to reveal its weaknesses.

- Finally, the adopt ring, to quote Mike Mason, an Agile Architect, is for when technologies are so good that "I will make fun of you at the pub if you aren't using them".

So, we built our own radar; here are the results with a brief description for some of the technologies. We hope it will serve you well!

Current Edition Highlights

- [Adopt] Azure Bot Service and LUIS
- [Adopt] Azure DevOps
- [Adopt] Mendix
- [Adopt] RxJS
- [Adopt] PowerBI
- [Trial] Azure Kubernetes Service
- [Trial] Kotlin
- [Trial] Progressive Web Apps
- [Trial] CosmosDB
- [Assess] Durable Functions in Azure Functions
- [Assess] Blazor
- [Assess] Vue.js
- [Assess] Flutter
- [Hold] Aurelia

This year our experts focused on 4 areas: .NET & Azure, Web & Mobile, Data & AI and Java & AWS. Technologies connected to machine learning emerged from the previous year and we have chosen to embrace them in our radar. We decided to connect Web and Mobile due to our focus on hybrid technologies.

.NET & Azure

- VSTS as CI/CD tool (1)
- Azure Bot Service and LUIS (2)
- Azure functions (3)
- .NET Core 2 (4)
- Immutable data objects for multi-threading apps (5)
- Docker & Docker Hub (6)
- Office 365 & Sharepoint (7)
- Azure AD (B2B / B2C) (8)
- SignalR as a Service (9)
- MS Flow / Logic Apps (10)
- Identity Server 4 (11)
- Azure Kubernetes Service (12)
- Service Fabric (13)
- Event Grid (14)
- Durable Functions in Azure Functions (15)
- Akka.NET (16)
- Blazor(17)

Web & Mobile

- Mendix (18)
- CI/CD (19)
- Modern web tooling (20)
- RxJS (21)
- Angular & React (22)
- TypeScript (23)
- Hybrid Mobile Applications - React Native & Ionic (24)
- Kendo UI (25)
- Bootstrap 4 (26)
- Firebase (27)
- Kotlin (28)

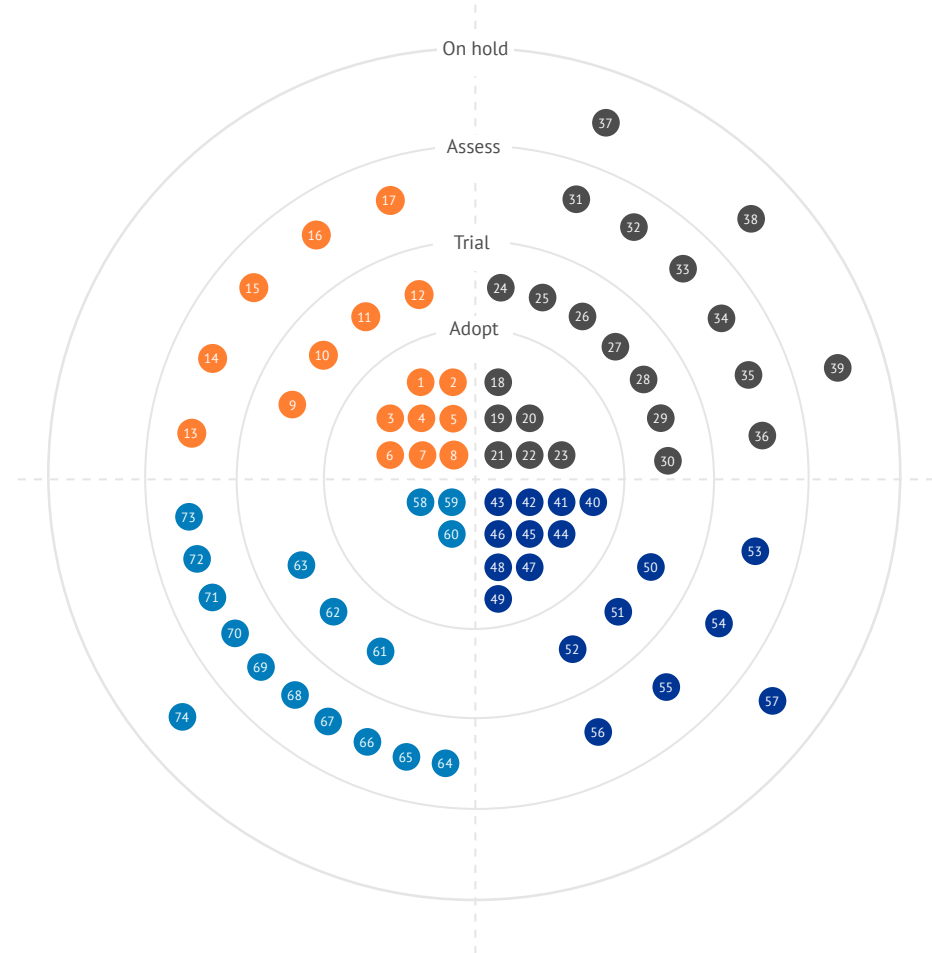
- Web Architecture Strategies (29)
- Progressive Web Apps (30)
- Web Assembly (31)
- Material Design (32)
- Vue.js (33)
- AR technologies (34)
- Flutter (35)
- Android JetPack (36)
- Aurelia (37)
- jQuery (38)
- Realm (39)

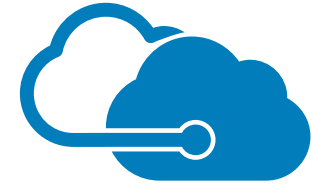
Data & AI

- MS SQL Server (40)
- PostgreSQL (41)
- Deep Learning (42)
- RoundhouseE (43)
- TDSP & CRISP-DM (44)
- Python (45)
- Elastic Stack (46)
- Tableau (47)
- R Language (48)
- Power BI (49)
- Azure SQL Data Warehouse (50)
- Azure CosmosDB (51)
- Big Data services (52)
- Azure Data Factory (53)
- Alteryx (54)
- Microsoft SQL Server 2017 Machine Learning Services (in Database & Standalone) (55)
- Microsoft Azure Machine Learning Studio (56)
- Teradata Database (57)

Java & AWS

- Spring (58)
- Amazon Elasticsearch Service (59)
- Lambda (60)
- Kinesis + Firehose (61)
- ECS (62)
- Aurora with PostgreSQL compatibility (63)
- AWS Autoscaling (64)
- Graal VM (65)
- Terraform (66)
- SageMaker (67)
- Reactive Programming for Java (68)
- Amazon Sumerian (AR & VR) (69)
- Amazon Elastic Container Service for Kubernetes - Amazon EKS (70)
- Alexa for business (71)
- Aurora Serverless (72)
- Kotlin (73)
- Oracle Forms (74)





.NET & Azure

Adopt Azure DevOps

Azure DevOps (formerly VSTS) is our current choice for CI/CD tool. It contains a Azure Pipelines - great amount of tools to build, test, and deploy your code to multiple different environments. Azure Repos allow to create Git repositories for your project and Azure Boards allows you to prepare backlog, team dashboards and pack all required project components in one place. Finally, Azure DevOps is free for small projects (up to five users), so you can easily try it in your team.

Adopt Azure Bot Service and LUIS

Azure Bot Services (ABS) is a framework to create an online chat-bot. It comes with various connectors and tools which allow us to respond to the most sophisticated needs. It has been proven in multiple projects, both for clients and internal, with great value delivered.

Adopt .NET Core 2

.NET Core has come a long way to become a stable and reliable framework.

For many months we were cautious with creating clients' applications in .NET Core due to the lack of functionalities, instability and feedback from our internal projects.

Currently .NET Core is our first choice for new projects/modules to ensure higher longevity, performance and portability of solutions. Cross-platform support

lets us host applications in stacks other than Microsoft's, which allows us to match our solution with diverse clients' needs. The framework comes with ASP.NET Core and Entity Framework Core - two foundations of web and database development. In recent months these libraries were quite leaky (e.g. lack of grouping in EF Core) but nowadays they are acceptable for starting new projects and rely on them.

Adopt Immutable data objects for multi-threading apps

Developers are used to working on classes which let them change their data whenever they need. This can be sufficient in single-thread processes but is very risky in multi-thread ones. That's why we recommend using immutable data structures in such environments. A different thread won't be able to change our data without clear information sent to other parts of the application. We achieve more visibility of our quicker bug identification.

Adopt Office 365 & Sharepoint

A few years ago Sharepoint was a tool which provoked hostile reactions among developers who were forced to work in it. Currently Sharepoint is a platform that is easy to develop in, with multiple options for frontend creation (Angular / React) and provides CI/CD experience. With connection to Office 365 you can integrate your bespoke application with an off-the-shelf solution to handle scenarios which would be time-consuming to implement on your own.

Adopt Azure AD (B2B / B2C)

Azure Active Directory is an identity management service. It takes care of hosting, resiliency, updating, and security for you. It has multiple options for handling authentication and access management. It allows us to shift repetitive work like user registration / login and focus on core areas of our business.

Trial SignalR as a Service

SignalR is a well-known standard to include real-time web functionalities in your application. It creates abstractions that allow to write your application faster and comprehensively handles connection management. For a long time we had to create applications to include SignalR functionalities. This isn't the case anymore!

SignalR Services allow you to use fully-managed services without worrying about hosting, scalability, load balancing or authentication. You can use serverless SignalR to incorporate it into your development process and build your application faster with less effort to maintain it.

Trial MS Flow / Logic Apps

Sometimes it is not necessary to write code to meet business needs. With Logic Apps you can design a client process as multiple connected blocks with multiple integration options. You can include conditional actions, parallel executions and error handling. You can't create very complicated scenarios in Logic Apps but for simple cases it's worth checking out.

Trial Identity Server 4

When Azure AD is not an option (e.g. you have to be cloud-agnostic), you still shouldn't write your own identity services - handling OAuth and OpenID properly isn't easy. In such cases we recommend using IdentityServer as a middleware to handle authentication. It is an open-source, reliable, mature and flexible, so you can easily adapt it to your own access control needs.

Trial Azure Kubernetes Service (AKS)

In recent years we've seen a boom in the use of Kubernetes. In our opinion, Kubernetes has won the battle between Mesos and Docker swarm and soon will be an open standard in container management systems. However hosting Kubernetes on your own is a tough job. That's why solutions like Azure Kubernetes Services are emerging to solve this problem. With AKS you can use Kubernetes in a managed environment without knowledge of all edge-cases involved in hosting this solution. It allows to kick off projects much faster and switch to on-premise Kubernetes when there is such a need later on.

Assess Service Fabric

Azure Service Fabric (ASF) is a platform which allows you to create distributed applications with an easy option to manage, deploy and package microservices applications. You can host Azure Service Fabric as a managed service in Azure, as well as in other clouds, on-premise data centers or on dev machines.

We have used ASF in few projects all of them were successful, but we're still trying to find best cases to include this technology in our architecture. Costs of introducing to ASF are high and XML configurations are complicated and hard to understand. Azure Service Fabric is definitely not a 'developer friendly' tool.

Assess Event Grid

A simple and powerful service to provide event delivery on a massive scale. It allows you to decouple publishers of events from subscribers to achieve scalable and reliable architecture. It stills lacks a few features but they are being implemented quite quickly (e.g. Cloud Events).

Assess Durable Functions in Azure Functions

Durable Functions are an option to connect multiple Azure Functions to create more sophisticated workflows. You can use multiple patterns like

- Function chaining
- Fan-out / fan-in
- Async HTTP APIs

There are some opinions that such usage of functions is a step back towards procedural programming, but it's worth trying if Durable Functions can be the orchestration for more complex scenarios.

Assess Akka.NET

Akka.NET is a big toolbox for building highly concurrent, distributed, and fault tolerant event-driven applications on .NET. It uses actor models to provide a higher level of abstraction for writing concurrent and distributed systems.

If you are building a digital twin, reacting to IoT events in rule-based manner, Akka.NET might be a core of your architecture.

Assess Blazor

Blazor is an experimental web framework where you can use C# and Razor in your browser. This is Microsoft's take on how potentially you might avoid rewriting your UI to newer MV* framework every three years, as with Blazor your team will be a group of self-sufficient full stack .NET developers. It uses WebAssembly to compile your code into runtime that can be run inside browser without any extensions or plugins. This technology is in a early phase but we look forward to giving it a try in POCs.

Web & Mobile



Adopt Mendix

Mendix is a high-productivity application platform. It helps us in building Mobile and Web apps not only by speeding the build phase but also by optimising the entire application lifecycle. With Mendix we are able to deliver not only simple forms-based apps but we can combine it with modern and sophisticated technologies like ML or IoT.

With this platform, we can deliver business value to our customers faster and with better quality. For them, it means that they can significantly reduce time to market of the developed solution.

Adopt CI/CD for Web

A few years ago mobile and frontend parts of applications were treated as inferior ones - developers and architects didn't focus on proper architecture or CI/CD for them. Currently we see a lot of great tools for providing great productivity and we recommend including them in every project regardless of size. For each case you can find the best-fitting platform to meet your needs, to name a few: VSTS / Travis CI / CircleCI / Nevercode / Bitrise. You can build your pipeline from the included blocks or define custom CI as a Service with YAML files.

Adopt Modern web tooling

Web development is changing rapidly and we need to keep up with the best tools suited to our and our clients' needs.

For long-running projects we recommend using Webpack - a module builder with predefined configurations for react / angular which allows to quickly kick off your projects. You can easily add functionalities like bundling, lazy-loading, transpiling, live-reloading etc. For shorter projects gulp or npm scripts might be sufficient

Whatever is your choice we consider npm to still be our 1st choice regarding package management (with huge performance boost in 5th and 6th version). E2E tests (with Express, cypress.io) and snapshot tests (with Jest) can be added when required.

Adopt RxJS

RxJS is a reactive extension for Javascript, based on the ReactiveX Observer pattern. We find it very useful to create adaptive and scalable solutions for frontend - to handle complex logic, retries or side-effects. With RxJS our applications just get simpler.

Adopt Angular & React

Angular & React are two different frontend libraries, but we've combined them into one - they are our first choice for frontend framework. They are stable and mature, with a lot of materials for starting and developing applications. You can easily combine them with different tools and languages like webpack, npm or Typescript. Support and community level is very high, you can get help with problems and proposed solutions to solve or evaluate. Both of them offer libraries allowing developers to test, create CI/CD and live reload.

Angular differs from React, but whichever you choose, you will be able to create a successful application.

Adopt TypeScript

TypeScript is our first choice as a language for frontend applications. TypeScript adds multiple features like optional types, classes, generics and modules to JavaScript. TypeScript compiles to readable, standards-based JavaScript with an option to debug in Typescript itself. Our projects written in Typescript are maintainable, refactorable and easier to understand, even after longer periods of time.

Trial Hybrid Mobile Applications - React Native & Ionic

Creating native applications in Android and iOS requires developers dedicated to every platform plus good understanding of related languages and tools. That's why Hybrid Mobile Applications emerged - they allow you to create one solution which can be run in both Android and iOS. If you are developing internal mobile app, you have web skills and time-to-market is more important than smoothness of UX or access to native hardware functions, you should try hybrid technologies.

In Hybrid Mobile Applications we recommend React Native and Ionic - both are battle-tested in our environments, with a lot of materials which we can rely on.

Trial Kendo UI

When you want to build a corporate-ready solution with multiple developers, randomly-picked UI controls can generate more harm than value. Their incoherent APIs will create confusion, causing your team to waste time. Moreover, styling in every control is different so you will need to spend your resources on creating UI guidelines.

In such a situation Kendo UI can help - it provides a range of powerful controls with uniform styling and API. With this toolset you can focus on delivering the same feeling in every page of your solution. Kendo UI is quite expensive and hard to extend but we assure you that when working with 15+ developers, Kendo UI can provide enormous time savings.

Trial Bootstrap 4

Bootstrap is a front-end framework (library) for designing websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Bootstrap also gives the ability to quickly prototype ideas or build entire apps with Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins.

Trial Firebase

Firebase is a mobile and web application development platform which delivers built-in handy tools for app development. Sample tools include: real-time online/offline database, Cloud Firestore NoSQL database, out of the box authentication through email, Google, Facebook, Twitter, Github etc, Cloud storage for photos/videos and other assets, Push notifications, Remote Config, Admob, Analytics, crash report analytics and more. <https://firebase.google.com/>

Trial Kotlin

Kotlin is a modern statically-typed programming language that runs on Java virtual machine and can also be compiled to JavaScript source code. Kotlin is a programming language fully supported by Google on Android and is directly included in the Android Studio 3.0 IDE package as an alternative to the standard Java compiler. <https://kotlinlang.org/>

Trial Web Architecture Strategies

Web applications do not consist of views for backend API operations anymore - more and more logic is being moved to frontend. You need to find a way to structure your applications to achieve scalability and maintainability. You have few strategies that allow you to achieve that goal:

- MVVM Architecture
- Redux / Reflux
- Reactive Architecture
- Micro UI - modularity / asynchronicity
- Domain Focused modules

Each can help you to reduce complexity and get better understanding.

Trial Progressive Web Apps (PWAs)

Progressive Web Apps are applications that behave like mobile apps while being developed like web apps. This type of application attempts to combine multiple features offered by modern browsers with the benefit of a mobile experience.

We have successfully delivered PWA for few clients with lower than for native application. With high popularity of web technologies, Google's support and increasing power of mobile processors, we expect this technology to become a standard quite fast.

Assess Web Assembly

Previously you could only use JavaScript in the browser - other languages were translated to JavaScript. WebAssembly provides a way to run code written in multiple languages on the web at near-native speed, with client apps running in the browser. There aren't a lot of materials about creating successful applications using WebAssembly, but we're placing a lot of hope in this solution.

Assess Material Design

Material Design is a design language developed in 2014 by Google. It provides multiple guidelines on designing applications with a similar feel to Google's services and tools.

Assess Vue.js

Vue is a very popular JavaScript frontend framework that's experiencing a huge growth. We find this library worth checking out PoC or some internal projects - there is a growing community and learning materials, but we have come across multiple scenarios that Vue couldn't really resolve. It is worth monitoring how this framework develops in the coming years.

Assess AR technologies

Augmented Reality (AR) technologies have emerged in the context of mobile devices. Currently, such devices are so powerful that they can handle sophisticated graphic rendering and animations. We've tried AR.js in a few internal projects with success, but unfortunately also with significant cost. There's a high interest from customers about AR solutions, but the technology is yet lacking cross-platform standards and maturity.

Assess Flutter

Flutter is Google's modern, reactive mobile app SDK for crafting high-quality native interfaces with native performance on iOS and Android in record time. It provides a fast development environment with "hot reload" feature. It has an expressive and flexible UI - built-in beautiful Material Design and Cupertino (iOS-flavor) widgets, rich motion APIs, smooth natural scrolling, and platform awareness. <https://flutter.io/>

Assess Android JetPack

Jetpack is a set of libraries, tools and architectural guidance to help make it quick and easy to build great Android apps. It provides common infrastructure code for navigation, pagination and persistence (and much more) so you can focus on what makes your app unique. <https://developer.android.com/jetpack>

Hold Aurelia

Aurelia is a javascript framework created by Rob Eisenberg - founder of Durandal library. We've done a few internal projects in Aurelia and we are quite satisfied by the results. Unfortunately, Aurelia hasn't got enough attention from the developer community and fell behind in the race of web frameworks. With a larger community and more learning materials, we find Angular and React to be more promising to focus on than Aurelia.

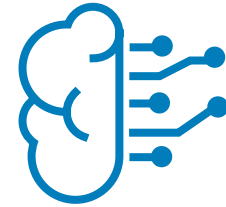
Hold jQuery

jQuery is a library broadly used in almost every web system. As of 2012, it is used by more than half of the Web's top sites. But we think that the time has come to leave jQuery and move to other solutions. For simple solutions you can use Vue or even VanillaJS, which in its current version is very reliable and effective.

Hold Realm

Realm is a database designed as a replacement for SQLite. For us it's deprecated as we encountered db-related crashes in the production.

Data & AI



Adopt MS SQL Server

Microsoft SQL Server (both on-premise and Azure version) is a foundation in our data storage and processing services. This very mature and well-investigated solution with multiple tools that answers our needs to provide best solutions. With connection to Reporting Services / Analysis Services / Integration Services it allows us to focus on delivery value regardless of which version we use in our client's project. MS SQL is actively developed - they have released JSON support, Machine Learning Services, graph support. To sum up - despite its age MS SQL is still the best option for providing data and visualisation services to your customers.

Adopt PostgreSQL

When we want / need to use a database outside of the Microsoft stack, PostgreSQL is our first choice. A mature, multi-model solution with a great community and variety of tools. Currently available in SaaS model in AWS and Azure clouds. We have used MySQL, MariaDB, and PostgreSQL and we think that this database is the most developer-friendly.

Adopt Deep Learning

Deep Learning can be provided using TensorFlow™ and Keras libraries from Python environment. TensorFlow™ is an open source software library for

high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano.

Adopt Roundhouse

Roundhouse (RH) is a database migrations engine that uses plain old SQL Scripts to transition a database from one version to another. Roundhouse currently works with Oracle, SQL Server, Access, MySQL, SQLite and PostgreSQL. It comes in the form of a tool, MSBuild, and an embeddable DLL. There is no visual tool at the current time.

RH is a SQL migration engine. It is not a migrations framework. You can use whatever migrations framework you want with Roundhouse as long as it can generate SQL files.

Adopt TDSP & CRISP-DM

The Microsoft TDSP Process and the CRISP-DM frameworks are frameworks for analytics projects that lead to standardized delivery for organizations, both large and small. CRISP-DM and TDSP focus on the business value and the results derived from analytics projects. The Cross Industry Standard Process for Data Mining or CRISP-DM model as it is known, is a process model that provides

a fluid framework for devising, creating, building, testing, and deploying machine learning solutions. The TDSP process model provides a dynamic framework to machine learning solutions that have been through a robust process of planning, producing, constructing, testing, and deploying models.

Adopt Python

Python is an interpreted high-level programming language for general-purpose programming. Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming in small and large scale. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python interpreters are available for many operating systems, is open source software and has a community-based development model. Python is the most popular language with many high-quality libraries and algorithms to provide data analysis and machine learning solutions.

Adopt Elastic Stack

Elastic Stack - Elasticsearch, Logstash, Kibana and Beats are four open source projects focused on storing, ingesting and visualising unstructured data sources. Elastic Stack makes searching and analyzing data easier than ever before. When you need to store enormous amount of data and query it successfully Elastic Stack should be considered at the top of the list.

Adopt Tableau

Tableau is a leading data visualization tool with multiple powerful features to provide great data visualisations and analysis. Tableau also comes with real-time data analytics capabilities and cloud support. We have implemented various reports and analysis in Tableau which weren't time consuming and provided a lot of insights for our clients.

Adopt Power BI

Power BI is a business analytics solution for visualizing data and sharing insights across an organization, or embedding them in applications and websites. Power BI allows you to connect to hundreds of data sources and bring data with live dashboards and reports. Power BI integrates with other tools, including Microsoft Excel.

Trial Azure SQL Data Warehouse

Sometimes you have to query petabytes of data - how to do this without worrying about the cost of creating such a platform for yourself? Azure SQL Data Warehouse offers a solution. It's an Enterprise Data Warehouse (EDW) that allows you to quickly run complex queries across massive amount of data.

Trial Azure CosmosDB

Use of NoSQL databases is still growing, mostly because of the ease of distributing data between nodes. That's why Microsoft came up with its own solution - Cosmos DB. Azure Cosmos DB is a globally-distributed, multi-model database service "for managing data at planet-scale." You can use CosmosDB as column / key-value / graph / document database to deliver your solutions faster and better without worrying about managing the nodes. Cosmos DB has its own limitations but even with them it's still a very good NoSQL database in the SaaS model.

Trial Big Data services

Performing Big Data analysis in cloud environments can be very effective. But you need the right tools to achieve it. That's why we recommend Apache Hadoop, Spark, and Hive. They allow processing enormous amount of data in real-time and integrates very well with the cloud services. As systems and devices "on the edge" start generating more and more data, analyzing them gives us valuable insights and helps in making the right decisions.

Assess Azure Data Factory

Azure Data Factory is Microsoft's cloud version of a data integration and orchestration tool. It allows you to move your data from one place to another and make data transformations. It contains multiple data source connectors which allow you to easily integrate different 3rd party systems into your applications.

Assess Alteryx

Alteryx is a combination of ETL(extract, transform, load), geospatial, and statistical modelling solution that promises to solve "getting-the-data-right" problem. It integrates very nicely with Tableau. Alteryx is an advanced data analytics platform intended to serve the needs of business analysts looking for a self-service solution. It contains 3 basic components: Gallery, Designer, and Server, which blend data from external sources and generate comprehensive reports. Each of them can be used separately.

Assess Microsoft SQL Server 2017 Machine Learning Services (in Database & Standalone)

SQL Server Machine Learning Services is an embedded, predictive analytics and data science engine that can execute R and Python code within a SQL Server database as stored procedures, as T-SQL script containing R or Python statements, or as R or Python code containing T-SQL. There are two options for using machine learning capabilities in SQL Server: In Database & Standalone. Support for each language is through proprietary Microsoft packages used for creating and training models of various types, scoring data, and parallel processing using the underlying system resources. Because the proprietary packages are built on open-source R and Python distributions, script or code that runs in SQL Server can also call base functions and use third-party packages compatible with the language version provided in SQL Server (Python 3.5 and recent versions of R, currently 3.3.3)

Assess Microsoft Azure Machine Learning Studio

Microsoft Azure Machine Learning Studio is a collaborative, drag-and-drop tool to build, test, and deploy predictive analytics solutions. Machine Learning Studio publishes models as web services that can be easily consumed by custom apps or BI tools such as Excel. Machine Learning Studio is a workbench environment that can be accessed through a web browser. Machine Learning Studio hosts a pallet of modules in a visual composition interface that helps to build an end-to-end, data-science workflow in the form of an experiment.

Hold Teradata Database

Teradata Database is an MPP (massively parallel processing) relational database system used to manage large data warehouses. It is produced by Teradata Corp. and delivered as a on premises server or Cloud solution. Teradata was used in one of our data warehouse solutions. It turned out that it performs poorly together with MS SQL Server Analysis Services cubes (processing of cubes is slower than fitted with MS SQL Server database). Taking that into account and the overall costs of the server and Teradata Services, it is not recommended to be widely used in Objectivity.

Big thanks to: Radosław Maziarka, Paweł Krawczyk, Rafał Szymczak, Jakub Kaczmarek, Piotr Zuzak, Piotr Gorzelany, Kamil Buczek, Jacek Kolonko, Patryk Lotzwi, Bartosz Hetmanski, Michał Jankowski, Edward Weinert, Magdalena Leszczyc-Laskowska, Krzysztof Kępa, Tomasz Bielaszewski, Konrad Łukasik, Michał Kozłowski, Tomasz Saluszewski, Damian Wolan, Grzegorz Strzelecki, Yarosław Gontar, Bartłomiej Horoniecki, Paweł Jabłoński, Ewelina Kiełkowska, Matthew La Fontaine

What do you think?

Do you agree with us?

Did we miss something?

Contact us through

Devs@objectivity.co.uk