# Глава 3. Изграждане на облачна система за производствено предприятие HeidelbergCement AG

Интро

## 3.1. Обща характеристика на дейността на компанията HeidelbergCement

HeidelbergCement е немска мултинационална компания за строителни материали със седалище в Хайделберг, Германия. След придобиване на 45% акционерно участие в Italcementi, HeidelbergCement става производител номер едно на строителни инертни материали, номер 2 в цимент и номер 3 бетон в световен мащаб. В Forbes Global 2000 за 2020 г. HeidelbergCement бе класирана като 678-ата най-голяма публична компания в света. Дейността на фирмата е в около 60 страни с 57 000 служители, работещи в 3 000 производствени обекта. HeidelbergCement управлява 139 циментови завода с годишен циментов капацитет от 176 милиона тона, повече от 1500 места за производство на готов бетон и над 600 кариери за инертни материали.

### 3.1.1. Основни бизнес процеси в компанията

Продуктите на компанията се използват за изграждане на къщи, инфраструктура, търговски и промишлени съоръжения, като по този начин отговарят на нуждите на нарастващото световно население за жилища, мобилност и икономическо развитие. Основната дейност включва производство и дистрибуция на цимент, инертни материали, готови бетонови смеси и асфалт.





### 3.1.2. Стимулиране на продажбите чрез цифрови технологии

Пакета от приложения стимулира прозрачността на данните, стандартизация в ERP, по-бързо и рентабилено планиране в заводите, създават иновативно решение, ориентирано към потребителите. Услугитe с добавена стойност, правят клиентите по-логистично интегрирани и по-добре оборудвани за да посрещнат предизвикателствата, свързани с устойчивостта. Целят да намалят материалните и логистични разходи, въглеродния отпечатък, да подобрят производителността и клиентския опит.

Според Dr. Dominik von Achten, председател на управителния съвет, дигитализацията е един от стълбовете на трансформацията, преминаващ през всички бизнес операции. Дигитални продукти, насочени към клиентите, целят да помогнат за успех в основния бизнес. Амбицията в дългосрочен план е да стане първата индустриална технологична компания в сектора.

Проследяване на камиона с готов бетон по пътя му към строителна площадка, съхраняване на билети и протоколи едно място, с мобилно и уеб приложение. Благодарение на контрола, циментът се произвежда по по-устойчив и ефективен начин, който намалява отпечатък върху околната среда и осигурява безопасност на работниците. Обслужването на клиенти се рационализира чрез постоянната връзка с бек-офиса и превозвача.

Технологичния пакет се основава на авангардни технологии с отворен код, с най-новите програми и езици. Контейнерни услуги, работещи изцяло в облак и разпределени в множество регионални клъстери.

## 3.2. Избор на технологични средства за разработка и операции

Тази подточка, ще се опише едно от най-важните решения, най-вече защото е почти необратимо. Основни съображения за изпълнение на задачата са общност в Stack Overflow, популярност според Google Trends и други.

Като метод за вземане на решене, базирано на данни, се представя анализ на разходите и ползите, който представлява процес на сравняване на прогнозираните или очакваните разходи и възможности (). Ако прогнозираните ползи надхвърлят разходите, можете да се твърди, че решението е добро. Показателите, използвани за измерване и сравнение са:

• Тип - статичен или динамичен;

• Зависим от флатформа/инфраструктура;

• Обюност;

• Производителност;

• Крива на обучение;

Таблици 3.3 представят анализ на сървърните технологии, подходящи за изпълнение на заданията.

***Таблица 3.3****: Сравнение на сървърни технологии за разработка.*



.NET e инструмент за разработка на софтуер, който Microsoft създава за собствената си екосистема от продукти и услуги. Той е стандарт, който разработчиците използват за създаване на софтуерни програми, които биват съвместими с технологичният продукти на Microsoft. Стартирането на .NET Framework се случва през 2002 г., с въвеждането на езикът за програмиране C#. Той предлага библиотеката с GUI настолни приложения, уеб рамката ASP.NET и функцията за достъп до данни ADO.NET. Цялата операция разчита на Common Language Runtime (CLR), тъй като това позволява компилирането и изпълнението на управлявания код. Има много причини, поради които .NET Framework и семейството .NET като цяло се използват, като например: обектно-ориентиран, система за кеширане, интегрирана среда за разработка на Visual Studio, универсални стандарти, общност, автоматичен мониторинг.

Въпреки всичко това, той идва и с доста неуспехи, като: проблеми с обектно-релационна поддръжка, тъй като съществуват опасения относно гъвкавостта на работната рамка по отношение на новите проекти на бази данни и тяхната поддръжка. Неприятен факт също е, че тъй като пакетът .NET е към Microsoft, всички промени или ограничения, които компанията може да наложи, неизбежно ще повлияят на проекти, изпълнявани под рамката. Също така и цената на лицензите.

Java е един от най-широко разпространените езици за програмиране. Причина за това е, че има много предимства, които помагат на програмистите да решават сложни проблеми от реалния свят. Съвместимостта на Java не зависи от операционната система или хардуера, което го прави независим от платформа. Подобно на .NET, Java е обектно-ориентиран език от високо ниво, осигуряващ автоматично освобождаване на паметта, многонишков с ефективна стратегия за разпределение на паметта.

Недостатъците на Java са сравнително слабата производителност, заемане на значителна част от паметта, платен търговски лиценз и други.

В днешно време хората широко използват езика PHP за създаване и разработване на уеб приложения. Сега той се превърна в един от основните езици за разработчиците, докато създават ново приложение. Водещи сайтове като Facebook и Харвардския университет използват езика PHP и го направиха по-популярен и надежден. Използването му се разви драстично през годините и сега хората го използват като прост инструмент за програмиране за разработване на уеб сървъри. Има много добродетели или предимства и няколко недостатъка. Нека ги разгледаме систематично. Някои от най-важните предимства на PHP са следните: oтворен код, независимост от платформа, лесно зареждане, удобен за потребителя, стабилен, връзка с база данни, поддръжка на библиотека и други. Въпреки че PHP е фантастичен език за програмиране, той има някои недостатъци като: проблеми със сигурността, лоша производителност, не подходящ за разработка на гигантски приложения, обработката на грешките и други.

Node.js е сървърно приложение като Apache, IIS, TOM и д.р, което изпълнява JavaScript. Самият Node не е изцяло написан на JavaScript, а основно на C. Node е платформа с отворен код за създаване на мрежови приложения в реално време, предоставя асинхронни, управлявани от събития входно/изходни операции. Плюсовете са: асинхронното управление помага за едновременното обработване на заявки, споделя база с код както на сървъра, така и на клиента, активна общност и други. Минусите му са, че Node.js не предоставя възможност за мащабиране, за да се възползва от множеството ядра, работа с релационни бази данни, не е подходящ за задачи, натоварващи процесора и други.

/python-pros-and-cons/

***Таблица 3.4****: Сравнение на мобилни технологии за разработка.*



## 3.4. Приложение на избраните технологии за изграждане на инфраструктурата в облачно базирана среда

По примери и указания от глава 2, тази подточка разглежда осъществяването на опростен във функционално отношение, облачен продукт, демонстриращ използването на .NET, Docker, Kubernetes в облачната среда на Microsoft Azure.

### 3.4.1 Хранилищата за данни в подсистемите

Различните back-end услуги, използвани от системата, имат различни изисквания за съхранение на данните. Azure предоставя много видове хранилища за данни, които могат да помогнат за поддръжка и извличане:

Azure SQL Database - Облачно базиран SQL Server. Поведението му е същото като това на основното изпълнение на базата, но предлага и много предимства: репликира в реално време данни в други географски региони, маскира данни за определени потребители, предоставя пълен одит на всички действия, които са се случили върху данните. Услугата е използвана от подсистемите за удостоверяване и каталогът за продуктите.

Azure Cosmos DB е нов вид нерелационна база данни, която работи с механизъм за съхранение и предоставяне на данни, който използва свободен модел, също така включва ниска латентност, репликация на данни в други географски региони в реално време, управление на трафика, автоматично индексиране на данните. Услугата е използвана от маркетинговата част.

Azure Blob представлява хранилище за съхраняване на големи неструктурирани данни. Това могат да бъдат фактури, изображения, видео, файлове и други. Услугата е използвана от подсистемата за поръчки.

Допълнение, Azure предоставя услуги за бази данни MySQL, PostgreSQL и MariaDB като универсално достъпни, мащабируеми, силно защитени и напълно управлявани.

Azure предоставя две хранилища за данни, които са много подходящи за съхранение на големи количества с цел анализ: Data Warehouse & Data Lake.

На фиг. 6 са показани различните услуги според структурата и характеристиките на данните.

Scatter chart

Description automatically generated with medium confidence

***Фиг. 6. Показва коя услуга за данни да се използва при определен сценарий***

### 3.4.2. Софтуерното внедряване и поддръжка в облачна среда

Най-използваната и наложила се като стандарт технология е **Docker**. Това е проект с отворен код за автоматизиране на внедряването на приложения като преносими, самодостатъчни контейнери, които могат да работят локално или в облака. Също така е компания, която популяризира и развива тази технология. Docker контейнерите могат да работят върху Linux или Windows. Предимства за разработчиците са: ускорено въвеждане на нови програмисти в проекта, премахнете конфликтите в приложенията, актуализиране и мигриране на софтуера.

На фиг. 7 е представено сравнение между виртуална машина и Docker контейнер.

Graphical user interface

Description automatically generated

***Фиг. 7. Виртуални машини и Docker контейнерите***

Виртуалните машини включват приложението, необходимите библиотеки и пълна операционна система. Изисква пълна виртуализация повече ресурси, повече време за стартиране в сравнение.

Докер контейнерите включват приложението и всички негови зависимости. Те обаче споделят ядрото на ОС с други контейнери, изпълняващи се като изолирани процеси в потребителското пространство на хост операционната система. (с изключение на Hyper-V контейнери, където всеки контейнер работи вътре в специална виртуална машина).

Виртуалните машини имат три основни слоя: инфраструктура, хост, операционна система, Hypervisor и всички необходими библиотеки. Слоевете в Docker са инфраструктурата, ОС и двигател за контейнери, който поддържа изолация, но споделя основните услуги на ОС. Тъй като контейнерите изискват много по-малко ресурси (например не се нуждаят от пълна ОС), те са лесни за изпълнение, внедряване и започват бързо. Основната цел на изображението е да направи зависимостите еднакви в различните среди. Това гарантирана еднакво поведение на всички среди: локална среда, среда за разработка или продуктивна.

Azure предоставя услуги, които могат да помогнат за постигане на много неща, варирайки от обикновени, като създаване на ново приложение с база от данни – до по-развити  като създаване на работни потоци за непрекъсната интеграция (CI) и внедряване (CD). Това са само няколко примера за някои често срещани работни похвати. Много от тях трябва да бъдат създадени индивидуално, но облачната инфраструктура предлага  всичко това като услуги. Силата на облака е, че ресурсите са невероятно устойчиви, малко вероятно е аварийно да спрат работа, тъй като центровете за данни са разположени по целия свят, състоящи се от десетки хиляди сървъри. Ако един сървър се повреди, друг поема управлението. Един от най-убедителните аргументи в полза на облака е, че може да разширява мащаба на услуги и ресурси почти безкрайно, в определени моменти, като например "Черен Петък" или голяма маркетингова кампания с промоции и намаления на артикули. Също така, когато натоварването намалее, мащабът може да се намали до обикновените си параметри. Уважавани и опитни облачни доставчици като Microsoft разпознават моделите на използване на нормалните потребители и тези на злонамерените. Инфраструктурата е предпазена от най-често срещаните атаки. Интелигентни инструменти за наблюдение, алгоритми за обучение и изкуственият интелект предоставят възможност да откриват атаки. При стартиране на приложения в Azure eдно от първите решения, които трябва бъдат вземети, са планираните за използване услуги:

• Azure App Services - eдин от най-лесните и мощни начини за хостване на приложения. Той е предпочитан при монолитната архитектура. Услугите са достъпни и работят в 99,95% от времето. Споделят мощни функции като автоматично мащабиране, внедряване с нулев застой и лесно удостоверяване, позволяват отстраняването на грешки в приложението докато работи в производствена среда (със Snapshot Debugger). По подразбиране приложението ще бъде достъпно в интернет, без да е необходимо да се настройва име на домейн или да се конфигурира DNS. Работи много добре с контейнери.

• Azure Virtual Machines - позволява преместване на съществуващи приложения от виртуални машини, които вече се изпълняват във център за данни. Има много предварително дефинирани изображения, които могат да бъдат използвани като Windows Server, който работи с IIS и има инсталиран и предварително конфигуриран ASP.NET на него, както и собствени софтуерни лицензи (като за SQL Server). Услугата е подходяща за мигриране на т.нар. „наследена система“, която да бъде използва като подсистема или източник на данни.

Следната таблица представя услугите и техните най-чести случаи на употреба:

Chart, scatter chart

Description automatically generated

***Фиг. 8. Представя кои услуги на Azure са подходящи за различните типове.***