

Cloudová Řešení

UAI/612

Kontakt

Ondřej Urbánek

ondrej.urbanek@orchitech.cz

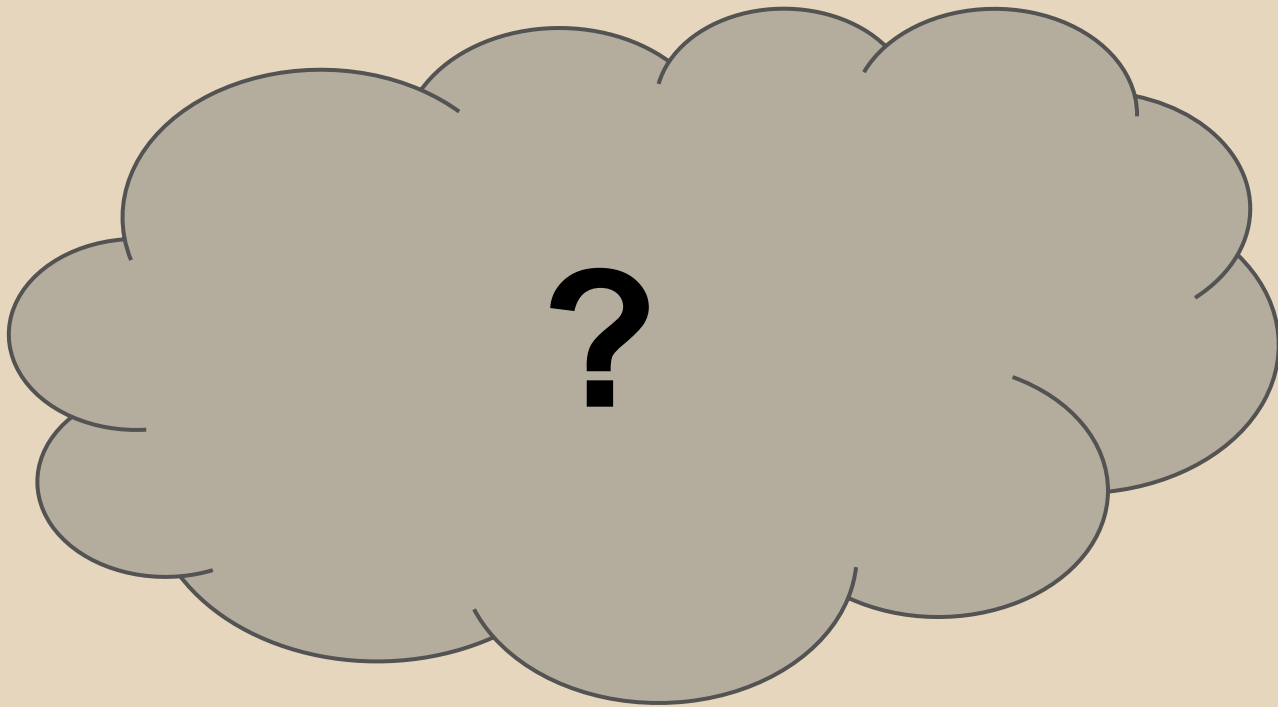
Výuka

- 7.3. 2014 13:00
- 21.3.2014 13:00
- 11.4. 2014 13:00
- 24.5. 2014 13:00

Cloudová Řešení

- Co je to cloud?
- Co je pro něj charakteristické?
- Jak lze cloudové řešení dělit?
- Proč cloud? Co přináší?
- Jaké využívá technologie?
- Jak vyvíjet aplikace pro cloud?
- Jaké jsou existující cloudová řešení?
- Jak postupovat při migraci do cloudu?

Cloud...



Cloud...

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Cloud...



Historie

- 1950s - Mainframe
- 1960 - John McCarthy
- 1970 - Virtual Machines
- 1990 - VPN Telekominikačních společností
- 2006 - Amazon WS
- 2008 - Eucalyptus, OpenNebula, OpenStack
- Platformy privátních cloudů

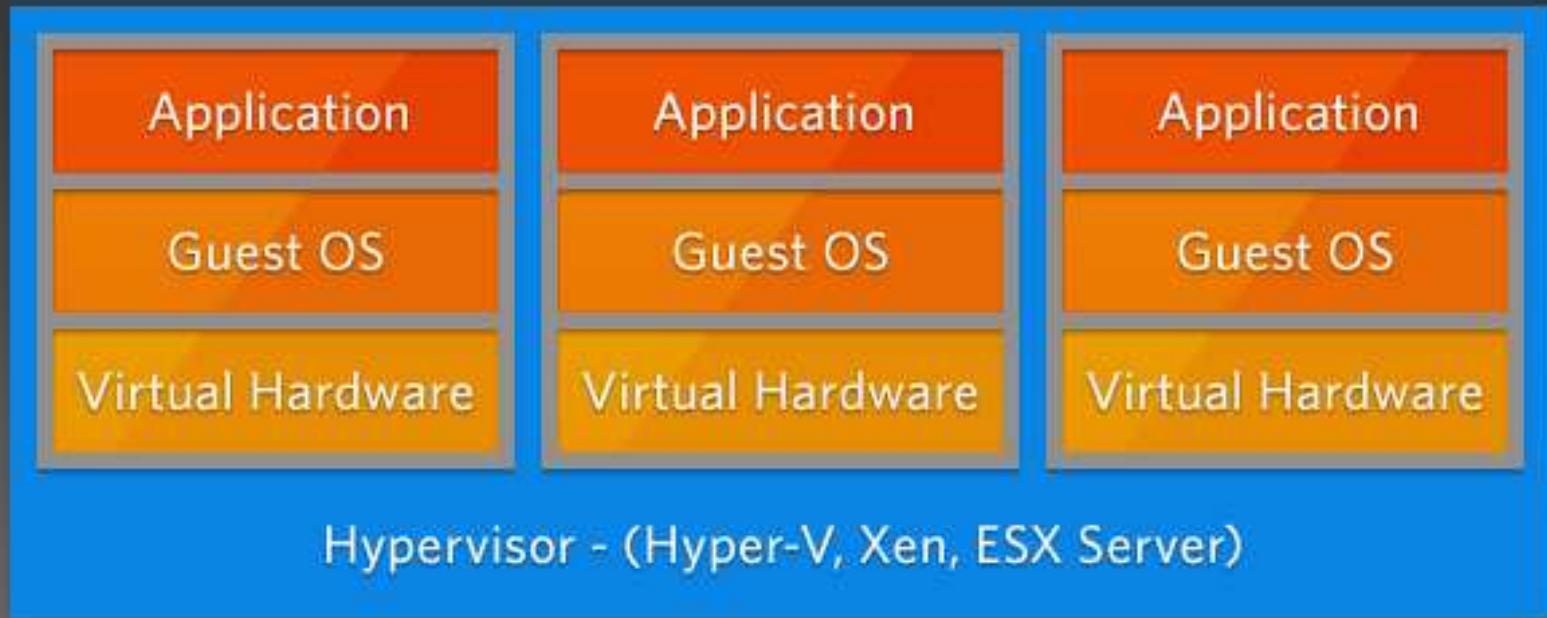
Vývoj cloudu

- Virtualizace, Hypervisor
- Přejchod od rozdělování výpočetního výkonu ke slučování
- Výstavba datových center
- Sdílení prostředků
- Přeprodej nevyužitého výkonu

Datová centra



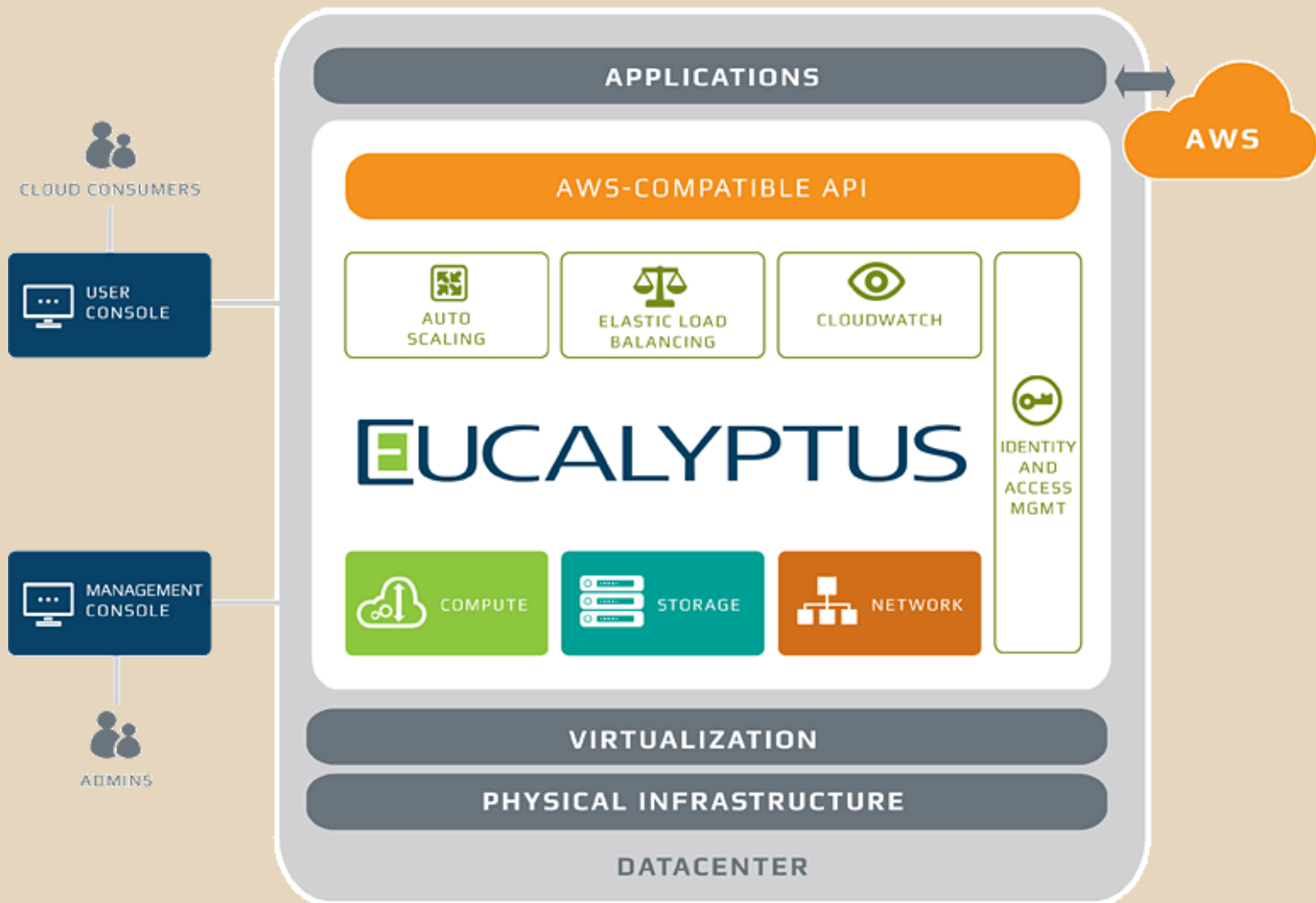
Virtualizace



Hardware - (CPU, Memory, NIC, Disk)



Cloudová architektura



Distribuční model

- Rozdělení na základě úrovně poskytovaných služeb
- Infrastructure as a service
- Platform as a service
- Software as a service
- Network as a service

IAAS

- Infrastructure as a Service
- Poskytuje základní výpočetní infrastrukturu
 - Virtuální stroje
 - Servery
 - Load ballancery
 - Sít'ovou infrastruktůru
 - ...

IaaS

- Google Compute Engine
- Windows Azure
- IBM Smart cloud
- Amazon EC2

PAAS

- Platform as a Service
- Poskytuje všechny prostředky pro běh aplikací a jejich dostupnost
- Automatické škálování, management zátěže aby byly splněny SLA
- Může obsahovat prostředky pro vývoj, testování a nasazení aplikací

PaaS

- Poskytuje základní výpočetní prvky

Operační systém

Běžové prostředí programovacích jazyků

Databáze

Webový server

...

- Microsoft Azure
- Google App Engine
- Heroku
- IBM Smart Cloud

SAAS

- Software as a Service
- Aplikace je poskytovaná jako služba dostupná přes internet (On-demand)
- Software a data jsou v cloudu (nezávisle na infrastruktuře)
- K aplikacím se přistupuje pomocí webového prohlížeče
- Aplikace je v neustálém vývoji
- Funkcionality jsou přidávány poskytovatelem transparentně k uživateli

SaaS

- Uživatel má omezené možnosti přizpůsobení
- Kompletně odstiňuje od infrastruktury
- Typicky se platí za počet uživatelů
 - Emailový klient
 - CRM, DMS,...
 - "Office" programy
- Google apps
- Microsoft Office 365

NaaS

- Network as a Service
- Poskytování konektivity
- Infrastruktura pro mezicloudovou komunikaci
- VPN
- Bandwidth on-demand

Distribuční model



SAAS

Software
as a Service



CONSUME



PAAS

Platform
as a Service



BUILD ON IT



IAAS

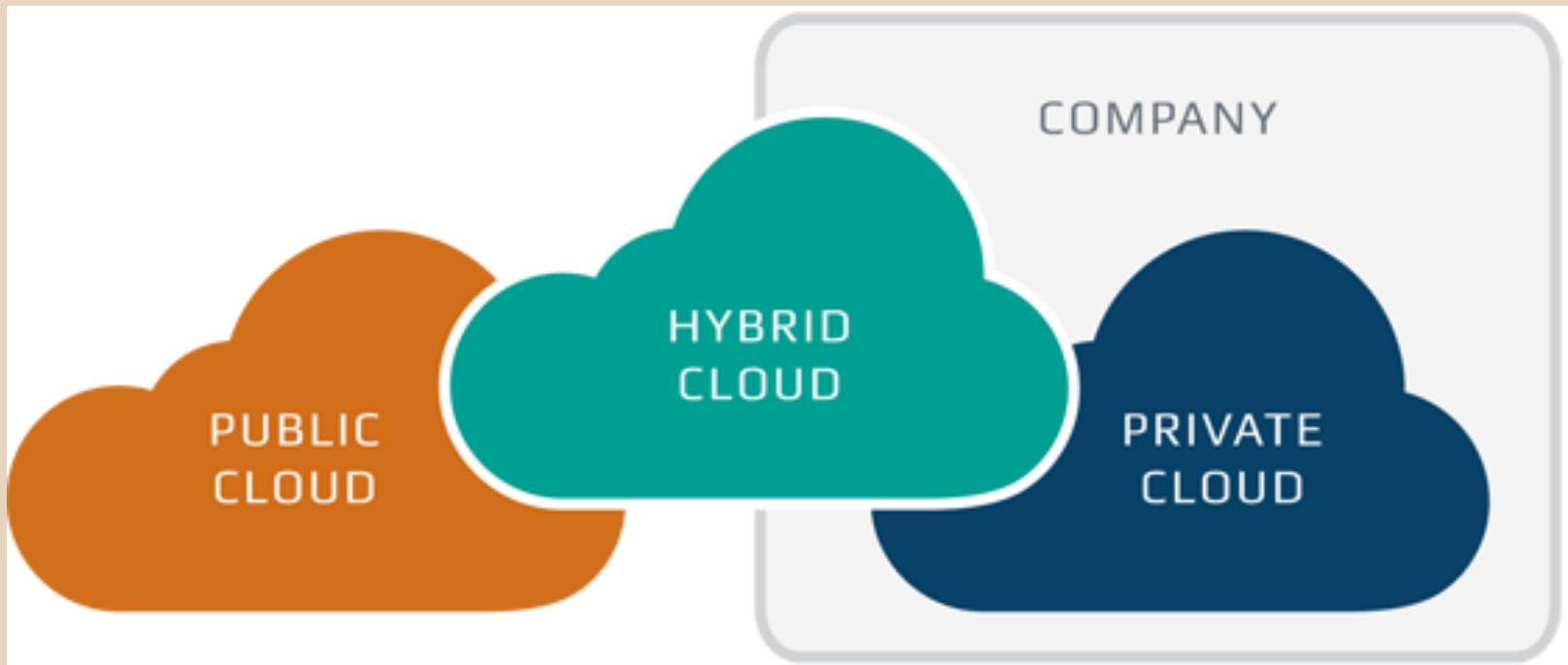
Infrastructure
as a Service



MIGRATE TO IT

Model nasazení

- Rozdělení podle místa provozu cloudové platformy



Veřejný cloud

- Poskytovaná služba je dostupná široké veřejnosti
- Většinou zdarma, případně zpoplatněno na základě používání
- Pro všechny uživatele ve stejné nebo podobné kvalitě
- Nízké počáteční náklady, definovaná cena provozu

Veřejný cloud

- Téměř neomezená škálovatelnost
- Finančně efektivní
- Spolehlivost
- Flexibilita
- Nezávislost na umístění
- Platba za využívání

Privátní cloud

- Cloudová infrastruktura vyhrazená pro konkrétní organizaci
- Provozována organizací samotnou, nebo poskytovatelem na vyhrazených zdrojích
- Upraven potřebám dané organizace
- Vysoké pořizovací náklady
- Zejména kvůli lokálnímu uložení dat

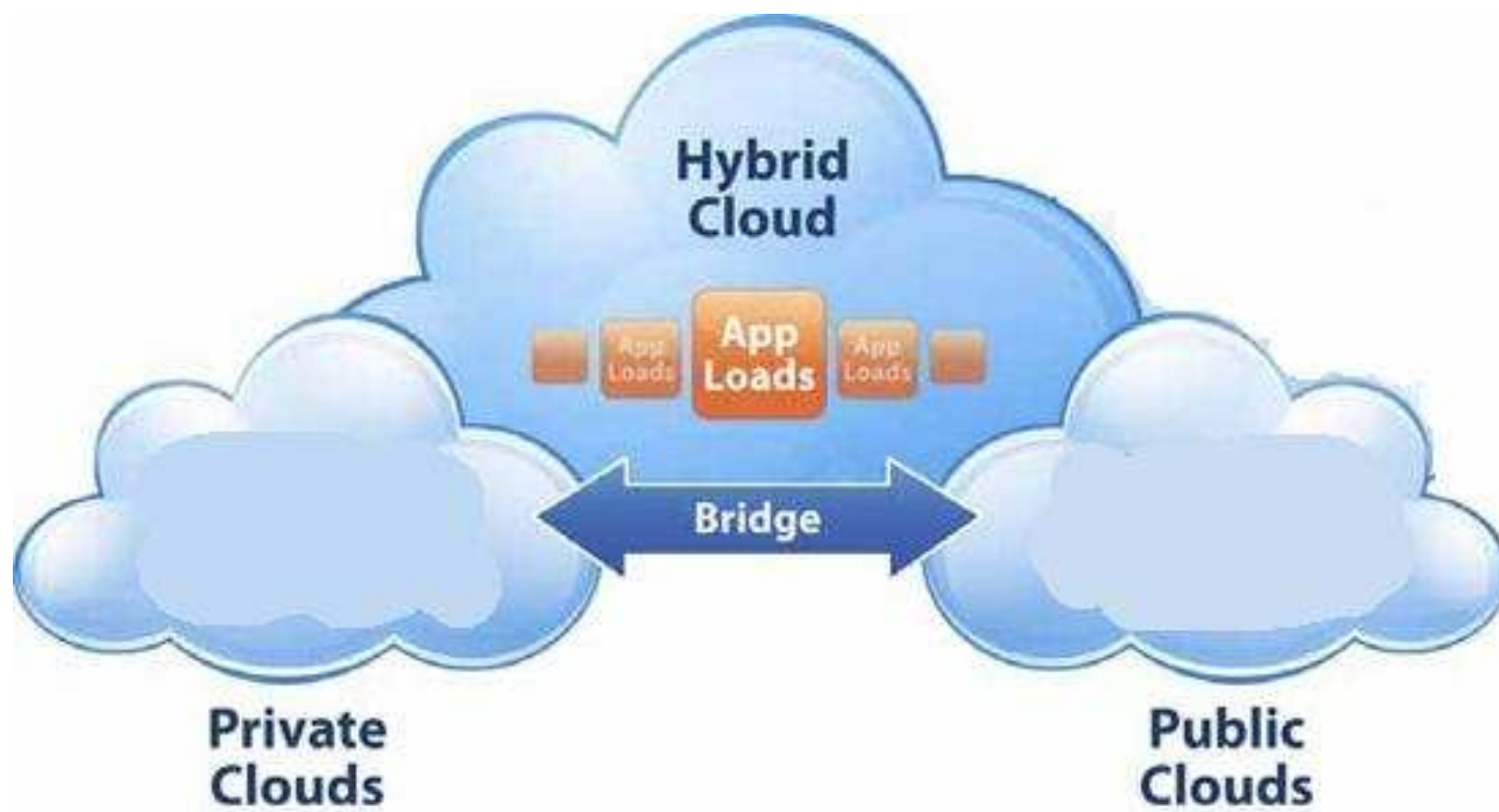
Privátní cloud

- Vysoká bezpečnost
- Větší kontrola
- Vysoká spolehlivost v rámci firmy

Hybridní cloud

- Vznikne propojením privátního a veřejného cloudu
- Navenek vystupují jako jeden cloud
- Propojení pomocí určených technologií
- Zlepšení škálovatelnosti privátních cloudů
- Cloud bursting

Hybridní cloud



Komunitní cloud

- Sdílení infrastruktury mezi více organizacemi v rámci nějaké komunity
- Stejná bezpečnostní politika, stejný předmět zájmu, ...
- Především jde o snížení nákladů

Virtuální privátní cloud

- Privátní cloud existující v rámci veřejného
- Pevně definované množství zdrojů které je kompletně k dispozici pro konkrétní organizaci
- Kompletně izolováno od ostatních uživatelů

Charakteristika



Nezávislost na umístění

- Aplikace jsou dostupné přes internet téměř odkudkoliv
- Cloud může být distribuován po různých částech světa
- Částečně lze umístění ovlivnit (např. země, datové centrum,...)

Samospráva "on demand"

- Uživatel si může kdykoliv nastavit potřebné parametry služby
 - Počet uživatelů
 - Velikost uložště
 - Počet procesorů
- Může probíhat bez zásahu administrátora

Spolehlivost

- Téměř 100% dostupnost
- Redundantnost zdrojů
- Automatické zálohování
- Cloudové systémy běží na velkých clusterech
- Automatický failover

Škálovatelnost

- Požadavky na výpočetní výkon se mění
- Cloudové systémy jsou schopny velmi pružně reagovat na výkonostní potřeby
- Snadné a téměř okamžité přidání zdrojů
- Autoscaling / elastic load ballancing

Multitenantnost

- Umožňuje sdílení stejné instance aplikace více organizacemi
- Uživatelé každé organizace musí mít přístup pouze ke svým datům
- Každý tenant může mít customizované rozhraní

Měření a monitorování

- Všechny parametry služeb lze sledovat
- Využití procesorů, uložistě, šířky pásma, počet uživatelů, ...

Bezpečnost

- Často vylepšená protože poskytovatel si může dovolit lepší ochrany než by si většina zákazníků mohla dovolit.
- Diskutabilní

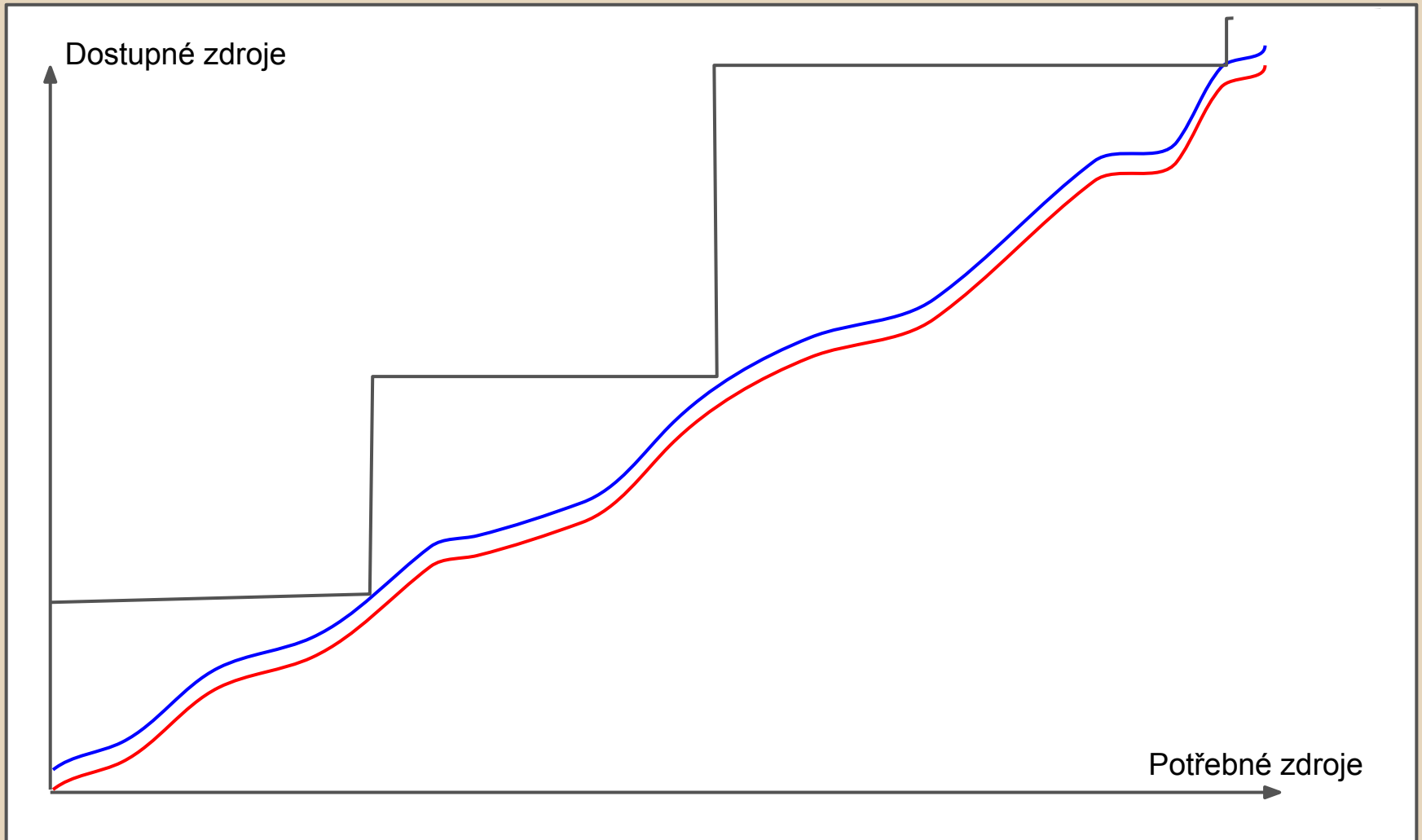
Motivace



Úspora nákladů

- Minimální počáteční investice / fixní náklady
- Platí se jen za to co je v danou chvíli potřeba
- Sdílení fyzických zdrojů více zákazníky (resource pooling)
- Úspora za provoz a administraci

Pay-as-you-grow



Pay-as-you-go

- Platíte jen za to co skutečně v danou chvíli spotřebujete
- Není potřeba plánovat rezervy
- Lze dynamicky měnit počet potřebných zdrojů na základě zatížení

Katalyzátor inovací

- Malé podniky mohou konkurovat na globální úrovni
- S minimálními náklady lze nasadit produkty a konkurovat i velkým podnikům

SLA

- Service Layer Agreement
- Dohoda mezi poskytovatelem a zákazníkem o kvalitě a dostupnosti služby
- Bezpečnost, zálohování, uptime

Nevýhody

- Nelze si zvolit konkrétní software a verzi
- Migrace může být velmi náročná
- Závislost na internetovém připojení
- Diskutabilní ochrana soukromí a bezpečnost dat
- Možné rozdíly v právním řádu poskytovatele a klienta
- Poměrně nový koncept, chybějící standardy, horizí vendor lock-in, špatná interoperabilita a portabilita
- Nevhodný pokud je vyžadována velká

Hrozby pro cloud

- Zneužití cloudu ke zločinným účelům
- Nezabezpečené API
- Někdo zevnitř
- Problémy se sdílenou technologií
- Ztráta nebo únik dat
- Ukradení účtu

UAI/612 - Cloudová Řešení

Technologie

Rekapitulace

- Multitenance
- Bezestavovost
- Škálovatelnost
- Cachování
- Bezpečnost
- Způsoby nasazení

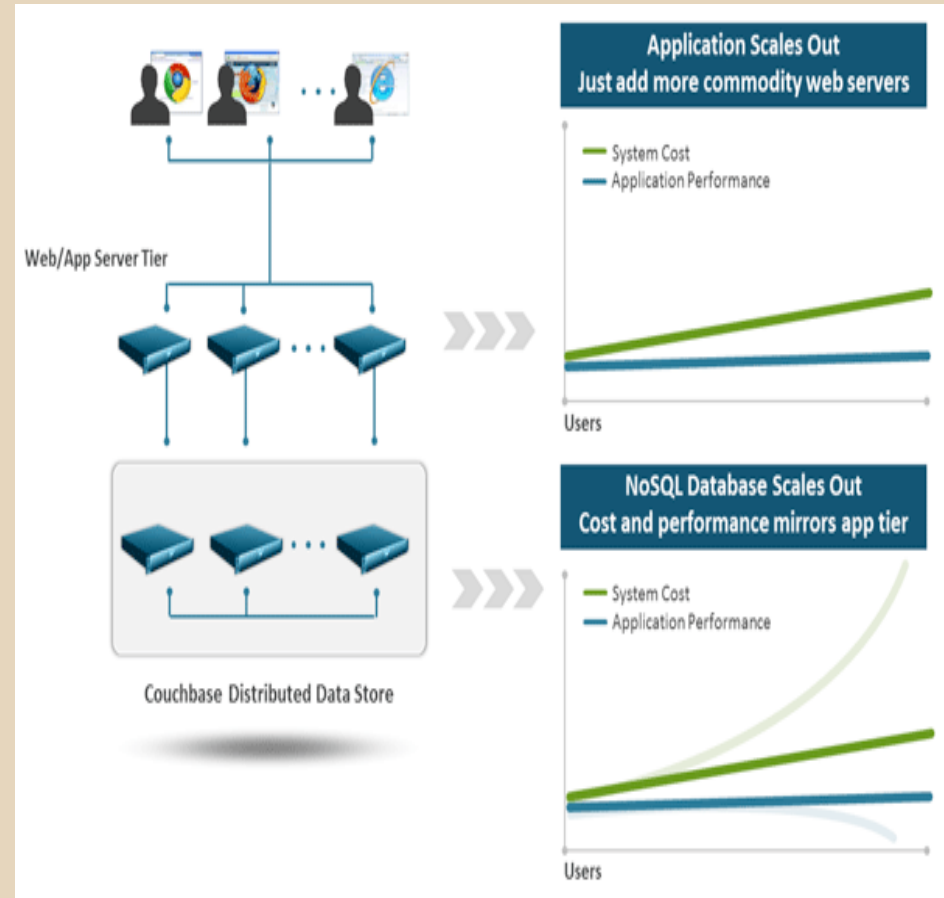
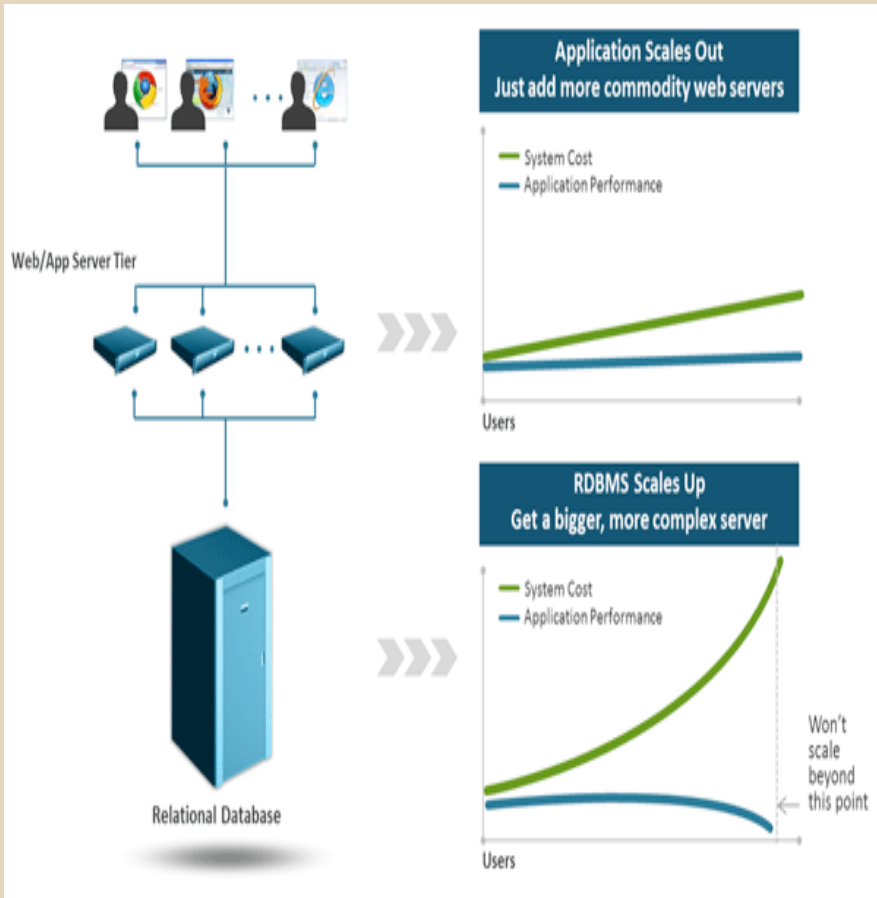
Datová úložiště

- SQL databáze
- NoSQL databáze
- Cloudová datová úložiště (API)
- Bloková úložště
- Archivační řešení
- Gateway

NoSQL Databáze

- Velmi dobře škálovatelné
- Založené na škálovatelných strukturách a architekturách
- Vytvořeny pro velké zatížení
- Autosharding
- Distribuované query
- Caching
- Nemají transakce a ACID

NoSQL Databáze



CDN

- Content Delivery Network
- Geograficky distribuovaná data



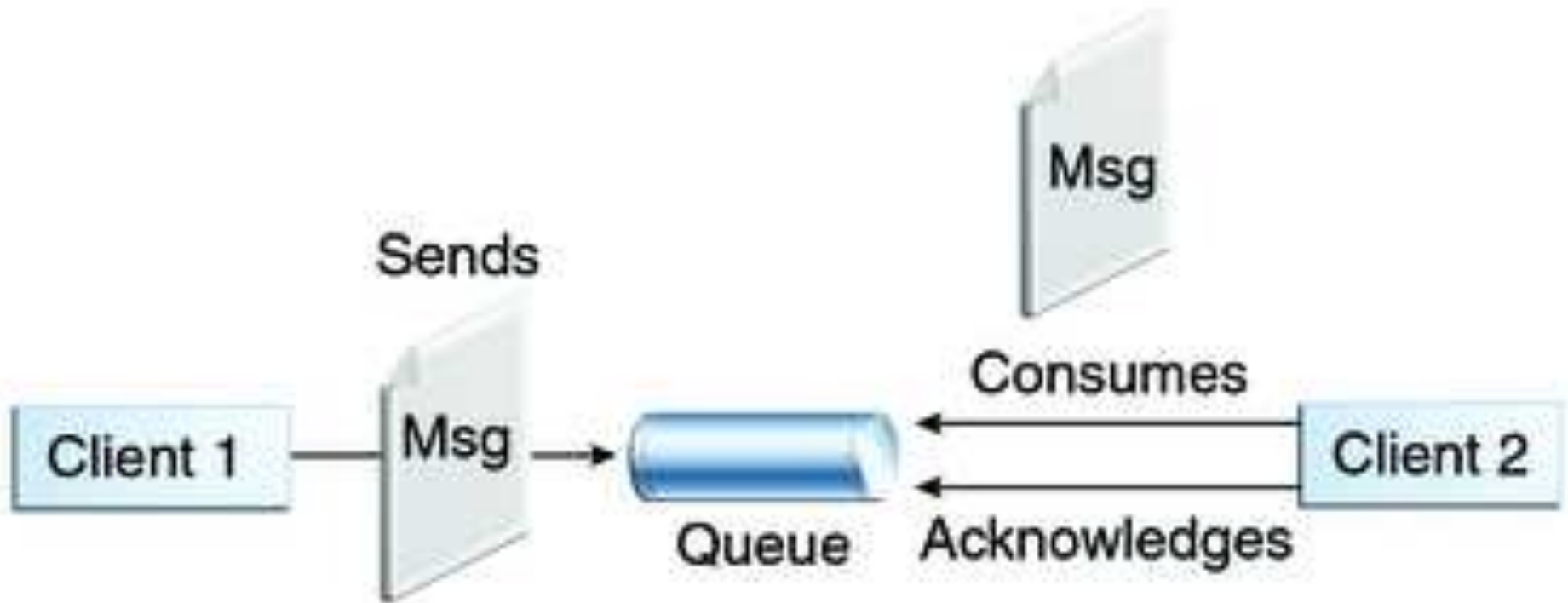
Komunikace

- Webové služby (middleware)
 - REST
 - SOAP
 - CORBA
- Messaging
- Push services

JMS

- API pro posílání zpráv
- Asynchronní zasílání zpráv
- Garance doručení zpráv
- Zpráva je doručena jednou a právě jednou
- Specifikace popisuje způsob jak vytvářet přijímat a zasílat a číst zprávy

Point-to-Point



Publish/Subscribe



Messaging

- JMS
 - ActiveMQ (Apache)
 - WebsphereMQ (IBM)
 - HornetQ (JBOS)
 - OpenMQ (SUN)
 - OpenJMS
- XMPP
- AMQP
- Proprietární cloudová řešení

XMPP

- Extensible Messaging and Presence Protocol
- IM, Presence information, Contact list maintenance, VoIP signaling, přenos dat..
- Decentralizovaný, Otevřený standard, Bezpečnost, Flexibilita
- Existující implementace v mnoha jazycích

Push services

- Google Cloud Messaging
- Apple push notification service

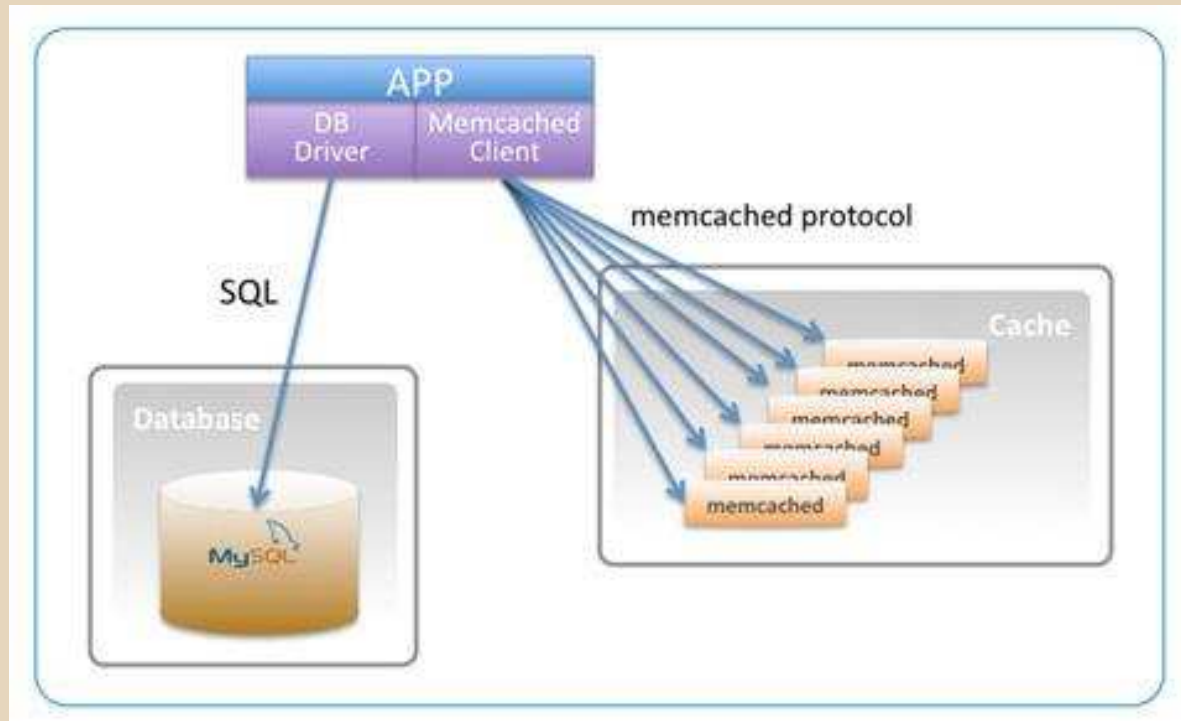
Cachování

- Vlastní implementace
- Cloudové řešení

Memcached

- Distribuovaný multiplatformní cachovací systém
- Běžící cachovací démon sloužící jako server
- Implementace klientů pro různé platformy
- Klient zná servery, stará se o failover
- Hashovací tabulka
- Klíč 250bytů, hodnota 1 megabyte

Memcached



Without Memcached

64MB Spare

web server

64MB Spare

web server

When Used Separately
Total Usable Cache size: **64MB**

With Memcached

Combined cache: 128MB

64MB spare

64MB spare

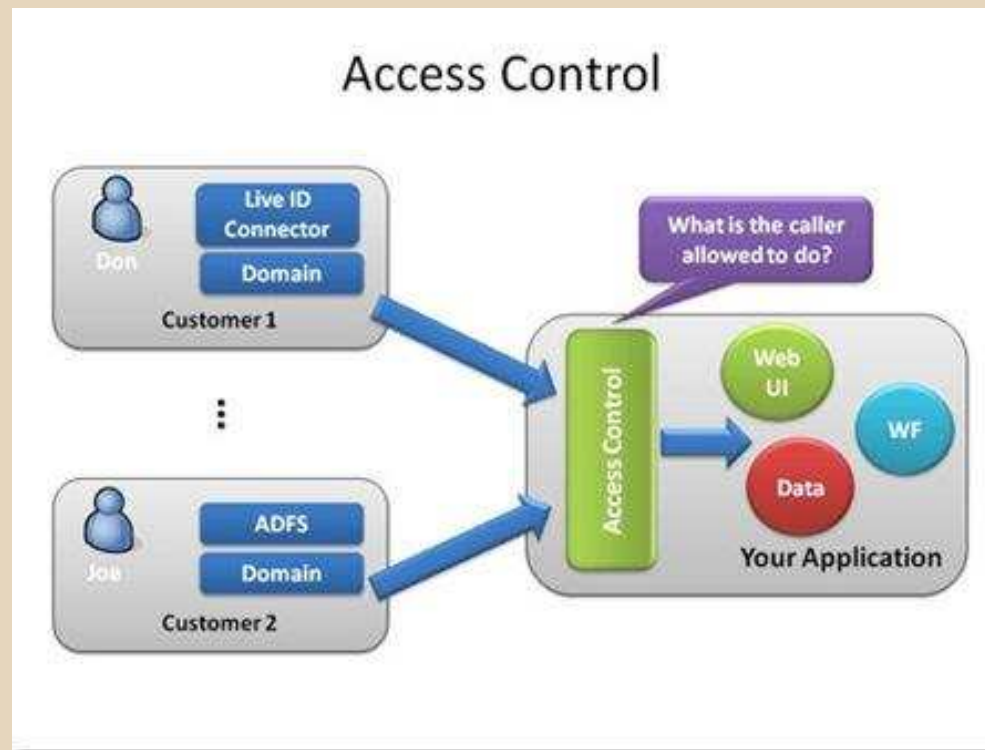
web
server

web
server

When Logically Combined
Total Usable Cache size: **128MB**

Řízení přístupu

- Vlastní implementace
- Cloudové řešení



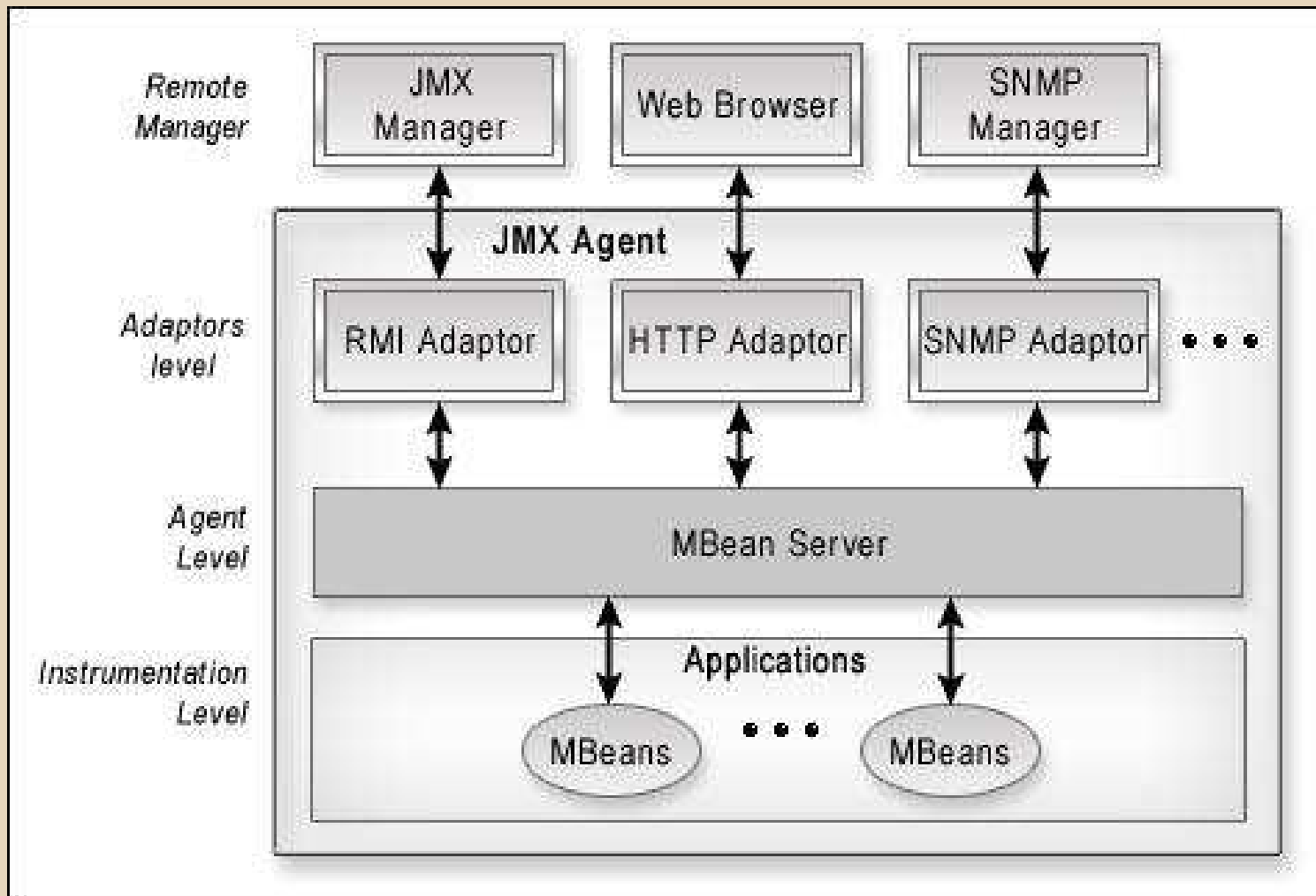
Management cloudu

- GUI
- Command line
- API
- Autoscaling

JMX

- Umožňuje management běžících aplikací
 - Např. změna konfigurace
- Umožňuje monitoring běžících aplikací
 - Monitoring stavu aplikace
 - Monitoring chyb, notifikaci o chybách
- Alerts, events, statistics
- MBean
 - Konfigurace aplikace
 - Modul programu
 - Identita Uživatele
 - Zařízení

JMX Architektura



Monitoring

- Monitoring vytížení instancí
- Monitoring sítě
- Monitoring uživatelských requestů
- Monitoring chyb
- Efektivita škálování

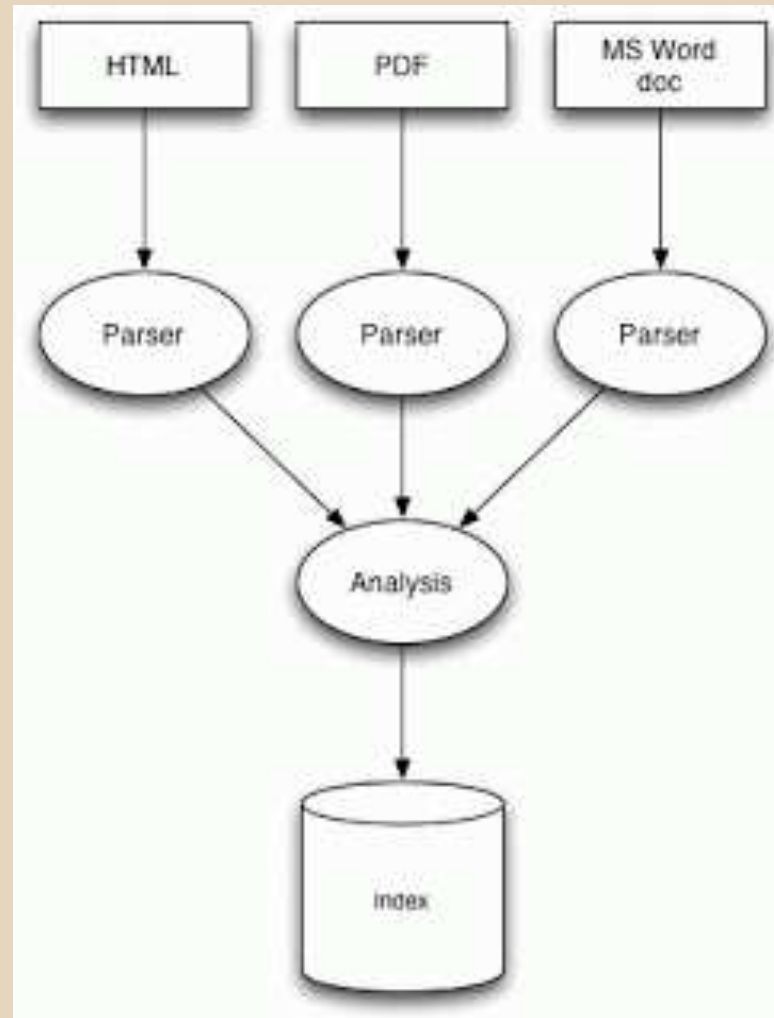
Indexace a fulltextové vyhledávání

- Vysoce výkonné vyhledávání ve velkém množství dat
- Provádění real-time analýza
- Faceted vyhledávání

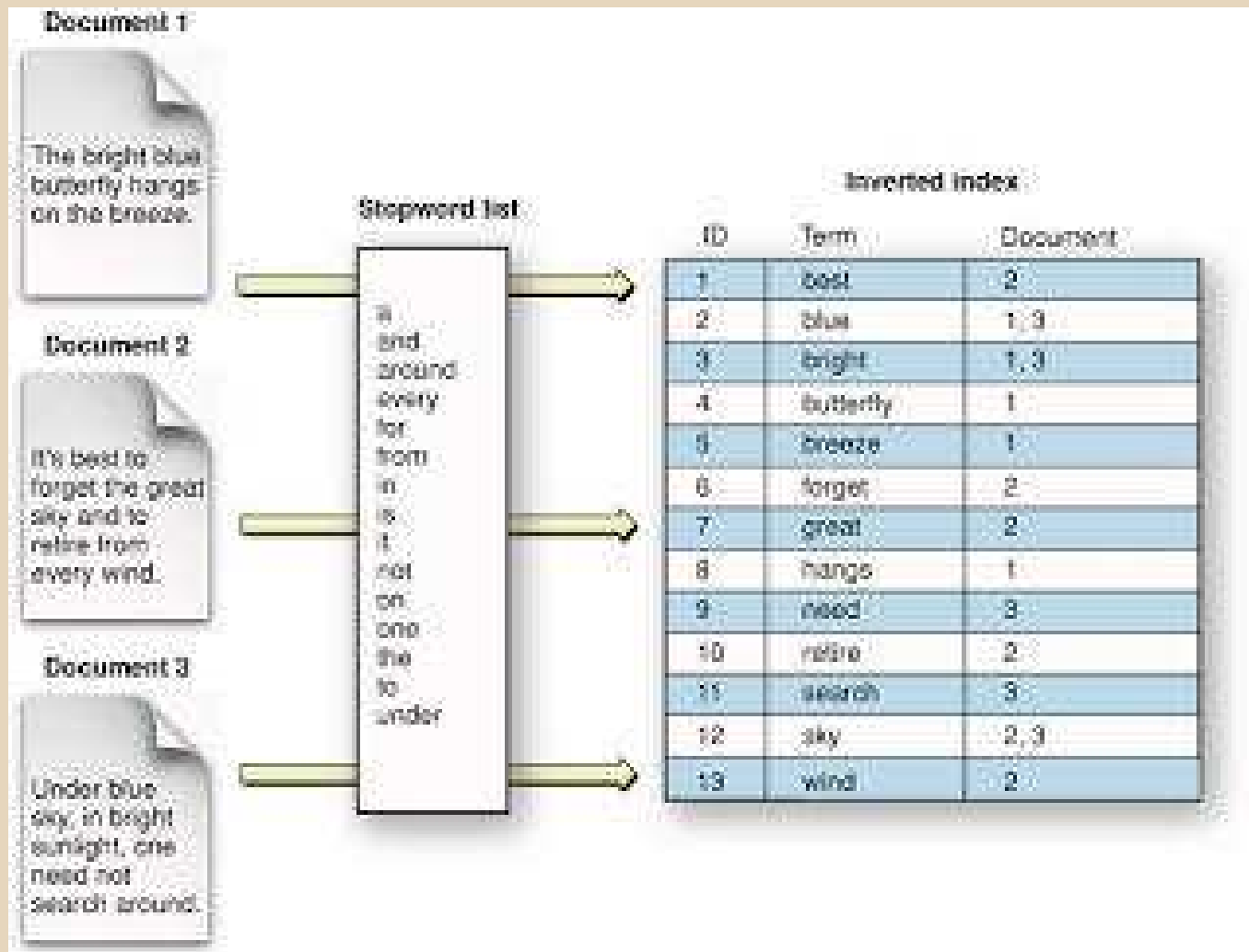
Lucene

- Fulltextové vyhledávání
- Indexace & Vyhledávání
- Napsáno v Javě, Portováno do mnoha dalších jazyků (index-kompatibilní)
- Inkrementální indexace, Batch Indexace
- Index asi 20-30% původního textu

Lucene



Lucene



Lucene

- Velmi výkoné vyhledávání
- Vyhledávání frází, Wildcard vyhledávání,
- Vyhledávání podle jednotlivých polí
- Řazení podle jednotlivých polí
- ...
- +include -exclude author:someone

ElasticSearch & SORL

- Indexovací a vyhledávací server
- Téměř Real-Time analýza dat
- RESTové rozhraní
- Založeno na Lucene
- Distribuovaný , snadno škálovatelný
- Podpora Multi-tenance
- Práce v clusteru
- Faceted search

Distribuované zpracování dat

- Cloud je ideální prostředí pro paralelizaci
- Možnost poměrně levně získat velký počet výpočetních prvků a velké datové uložště

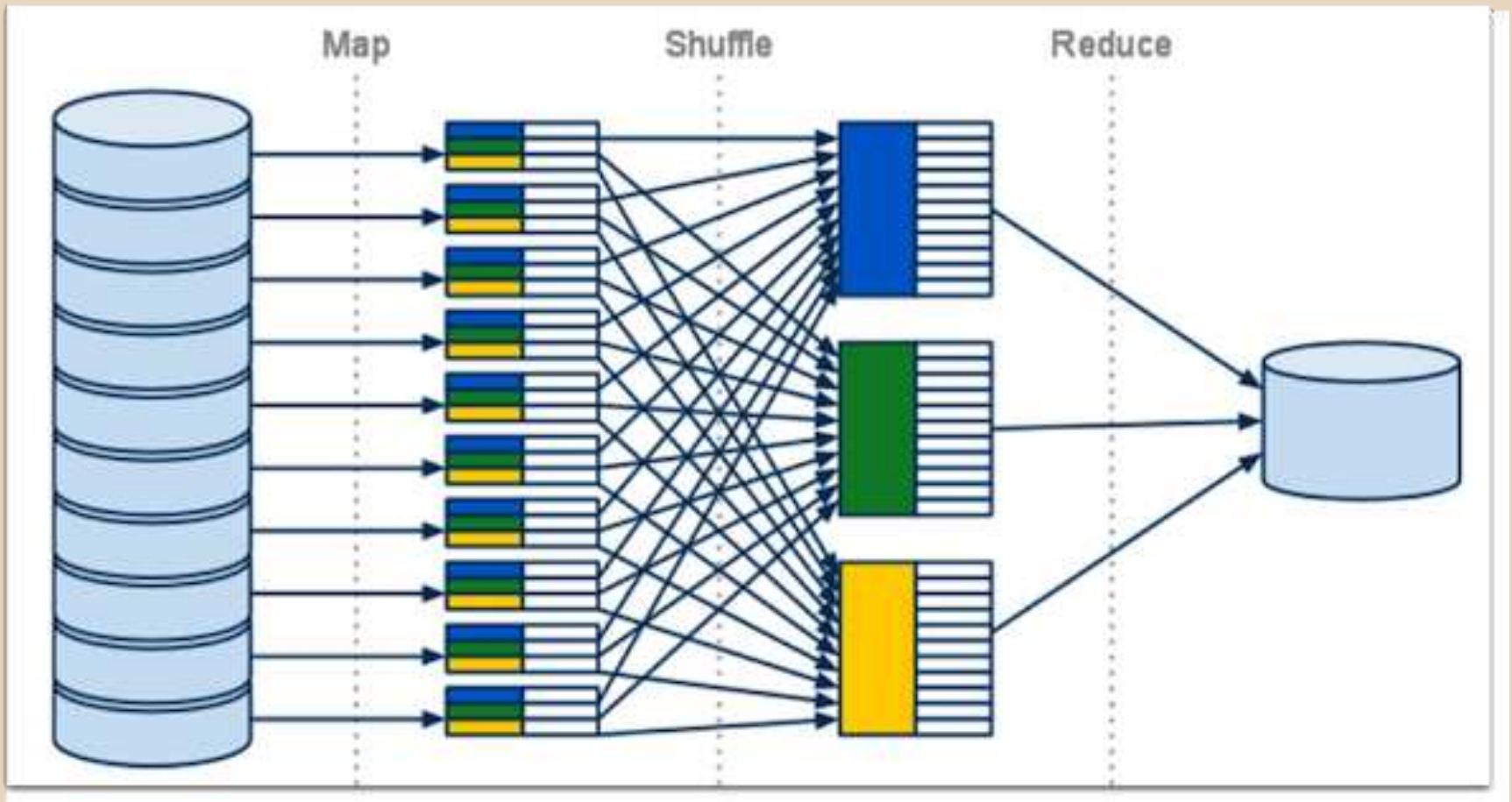
Hadoop

- Framework pro zpracování velkého množství nestrukturovaných distribuovaných dat
- Data uložena na velkém množství počítačů
- Navržený tak aby detekoval a zpracovával výpadky
- Výpočet probíhá paralelně na každém uzlu (blízko datům)
- Konečný výsledek sestaven z dílčích výsledků pomocí MapReduce

Hadoop

- Až 6000 uzlů (16 jader, 1000 úloh)
- Hadoop common
- Hadoop Distributed Filesystem
- Hadoop YARN
- Hadoop Map/Reduce
- Ambari

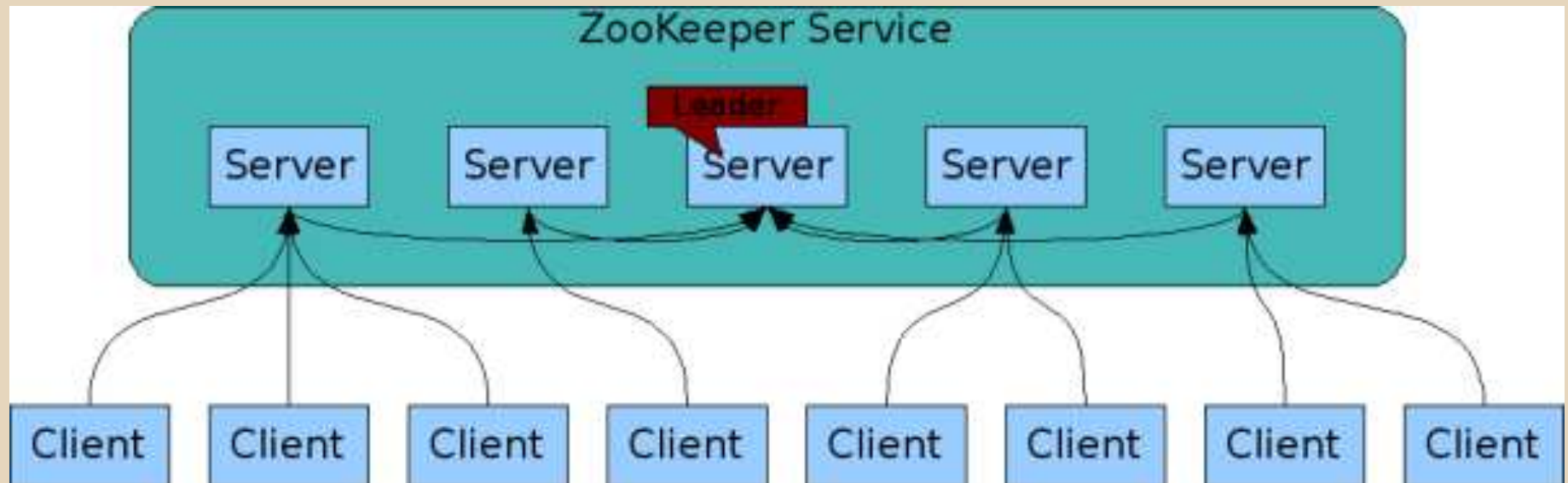
Map and Reduce



Zookeeper

- Distribuovaná synchronizace
- Správa konfigurace
- Koordinace distribuovaných procesů pomocí hierarchického datového úložiště
- Běží v paměti (rychlost, limitovaná velikost)
- Přístup k datům je uspořádaný

Zookeeper



Spring DATA

- Skupina projektů pro přístup k novým datovým úložištím
- Entity mapping, QueryDSL, ...
- REST, Redis, MongoDB, Hadoop, Solr, Elasticsearch

UAI/612 - Cloudová Řešení

Návrh aplikací pro cloud

Rekapitulace

- Cloud computing
- Virtualizace
- IaaS, PaaS, SaaS
- Veřejný, Privátní, Komunitní, Hybridní
- Motivace

Návrh aplikací pro cloud

- Software as a Service
- Multitenantní aplikace

- Návrh architektury
- Návrh datového uložště
- Návrh škálovatelnosti
- Model nasazení

Multitenance

- Architektura aplikace umožňující sdílení jedné aplikační instance více zákazníky (klienty, tenanty)
- Každý tenant má přístup jen ke svým datům
- Data tenantů jsou oddělena na základě designu aplikace a datového úložiště

Identifikace tenanta

- Na základě autentizace uživatele
- Na základě cesty v URL
- Na základě subdomény
- Na základě domény

Oddělení dat

- Oddělené databáze
- Sdílená databáze, oddělená schémata
- Sdílená databáze, oddělené tabulky
- Sdílená databáze, sdílené schéma

Přizpůsobitelnost

- Vzhled
 - Chování
 - Konfigurace
 - Vlastní kód
-
- Oddělená databáze
 - Sdílená databáze, rozdílné schema nebo tabulky
 - Fixní schema, fixní sloupce
 - Fixní schema, tabulka hodnot

Způsoby nasazení

- Multi instance, single tenant
 - Single instance, single tenant
 - Multi instance, multi tenant
-
- Rozdělení podle úrovně služby
 - Geografické umístění

Rozhodovací kritéria

- Izolace dat
 - Škálovatelnost
 - Výkon
 - Přizpůsobitelnost
-
- Počet tenantů
 - Počet uživatelů
 - Velikost DB
 - Geografické rozložení tenantů

Možné komplikace

- Číslování
- Řazení
- Výkon
- Bezpečnost dat
- Spouštění naplánovaných úkolů
- Volání webových služeb
- Identifikace tenanta mezi komponentami

Škálovatelnost a elasticita

- Schopnost systému efektivně se přizpůsobit aktuálním výkonovým potřebám
- Možnost přidání instancí a vypnutí instancí
- Odebráním instance, nebo přesměrováním uživatele na jinou instanci nedojde ke ztrátě uživatelských dat.

Volná vazba komponent (loose coupling)

- Míra znalosti jakou o sobě mají komponenty
- Komunikace pomocí definovaných rozhraní rozhraní
- Snižuje závislost na konkrétních technologiích
- Změn a v jedné komponentě je oddělená od zbytku aplikace
- Izolace problémů

Služby

- Autonomní jednotka realizující určitou vhodně zapouzdřenou činnost
- Poskytuje nějakou funkcionalitu pomocí pomocí definovaného rozhraní



Service Oriented Architecture (SOA)

- Principy a metody vývoje software ve formě interoperabilních distribuovaných služeb
- Aplikace se vytváří z na sobě nezávislých komponent poskytujících služby
- Komponenty lze znovupoužít k jiným účelům
- Jednotlivé služby mohou být implementovány na různých platformách
- Komunikace probíhá na základě dohodnutých rozhraní

Bezestavovost

- Stav není uložen v paměti aplikace
- Správa stavu zhoršuje škálovatelnost
- Při poruše je možné aplikaci vypnout a uživatele přesměrovat na jiný server
- Správa stavu na klientovi (cookies, URL)
- Uchovávání session na serveru zhoršuje load ballancing

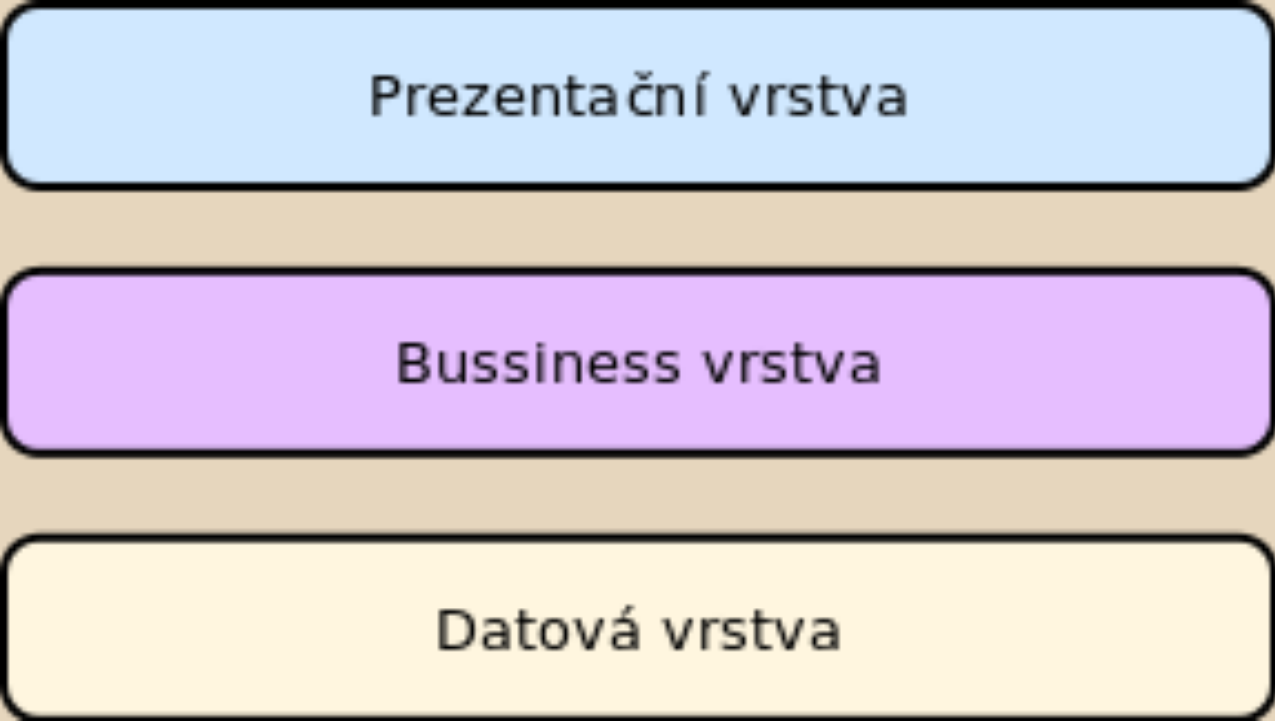
Správa session

- Na klientovi (JavaScript)
 - Klasická session (session storage)
 - Serializace a deserializace stavu
-
- Jednoduchost implementace
 - Cena
 - Výkon
 - Robustnost
 - User experience
 - Bezpečnost

Striktní oddělení dat a aplikační logiky

- Lepší flexibilita nasazení a škálování
- Nic není uloženo lokálně (včetně např. logů)
- Soubory, například mediální obrázky
- To umožňuje přidávat, odebírat nebo přemísťovat výpočetní jednotky podle
- Umožňuje transparentně měnit technologii ukládání dat

Třívrstvý design (3-tier design)



Prezentační vrstva

Bussiness vrstva

Datová vrstva

Komunikace komponent

- Pomocí middleware technologií
- Webové služby
- REST
- Messaging

REST

- Založeno na protokolu HTTP
- Orientováno kolem "resources"
- HTTP Metody (GET, PUT, POST, DELETE)
- Přenosový formát XML, JSON

Messaging

- Asynchronní komunikace pomocí zasílání zpráv
- Založeno na frontách zpráv
- Komunikace je nepřímá, zprostředkovaná další komponentou
- JMS

Publish - Subscribe Messaging

- Obdoba messaging
- Odběratelé se zaregistrují k určitému tématu
- Kdykoliv někdo publikuje zprávu k tématu je přeposlána všem odběratelům

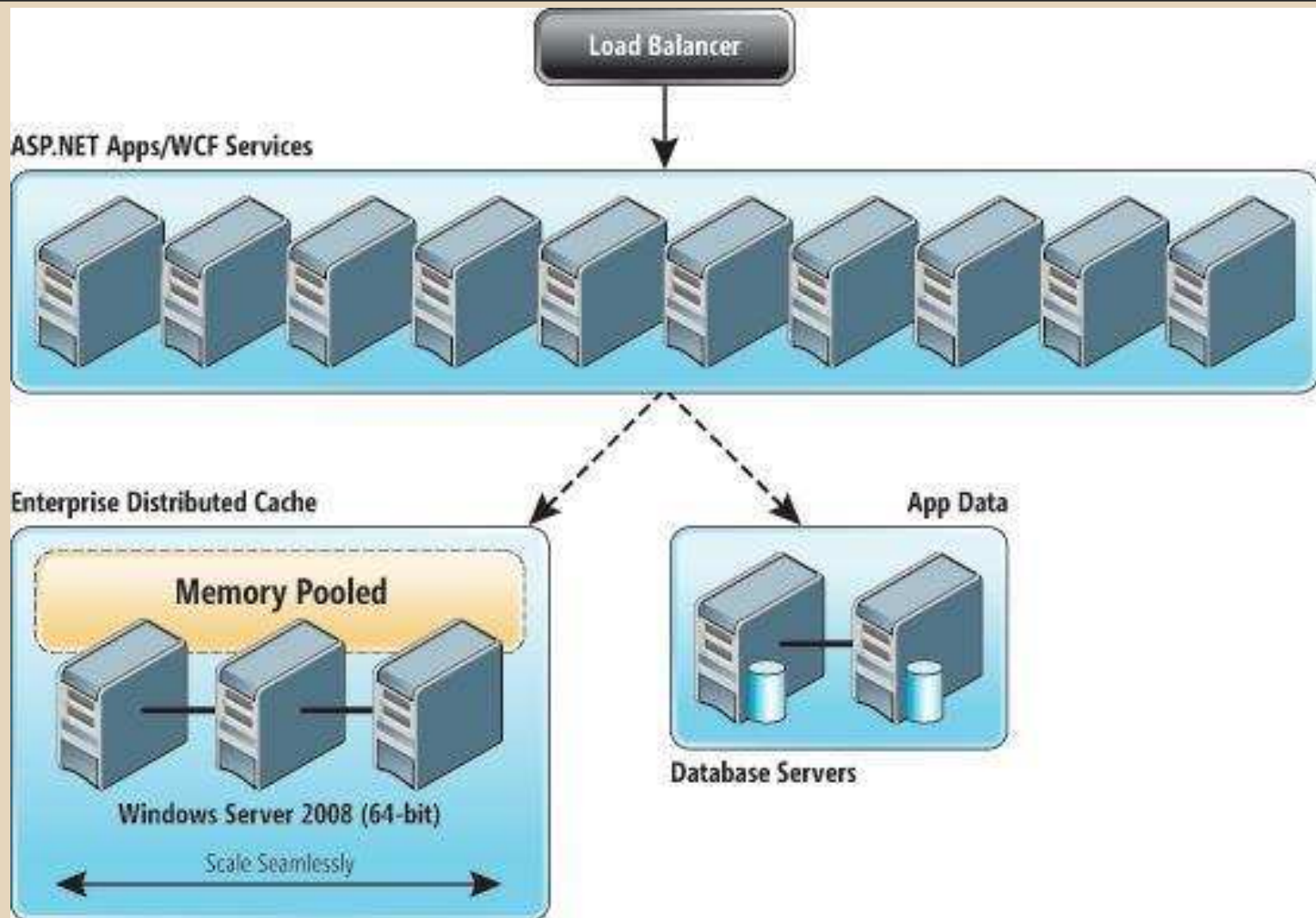
Cache

- Častá komunikace mezi službami může být pomalá
- Snižuje nároky na komunikace
- Zrychluje odezvy read operací
- Umožňuje částečnou funkci pokud není některá ze služeb dostupná
- Snížení latence

Distribuovaná cache

- Cache sdílená mezi více servery
- Cachovaná data jsou přenášena po síti
- Cache je rozprostřena mezi více strojů
- Navenek se tváří jako jediná cache
- Izolovaná cache, replikovaná cache, rozložená cache

Distribovaná cache



Nelze spoléhat na HW

- Všechny výpočetní jednotky jsou stejně (ne) spolehlivé
- Kterákoliv může kdykoliv selhat
- Nelze spoléhat že některá bude spolehlivější než jiná (např. load balancer)
- Vyvarujte se "single point of failure".
- Mean Time Of Error vs Mean Time Of Recovery

Self recovery design

- Vytvořte mechanismy pro automatické zotavení a rekonfiguraci cloudu
- Aplikace musí umožňovat bezobslužný start
- Health check

Fault tolerance

- Plánujte že některé zdroje nemusí být dostupné
- Komponenty by měly umět (dočasně) fungovat izolovaně
- Nemělo by dojít ke zhroucení celého systému
- Při opětovné dostupnosti zdroje se systém zotaví původního stavu
- Database throttling

Nelze spoléhat na síť

- Cloudové aplikace běží často v obrovských data centrech
- I pouhé dva počítače může dělit velké množství kabelů a switchů a nemusí být ani ve stejné budově
- Může docházet k výpadkům konektivity mezi dvěma uzly

Proprietární technologie

- Vendor lockin
- Nemožnost migrace na jiný typ cloudu

Neblokovat se

- Vyhnout se zamykání
- Uzamknutý resource znamená úzké hrdlo
- Pokud možno oddělit ukládání dat od jejich zpracování

Bezpečnost

- Důkladné zabezpečení na všech vrstvách
- V cloudovém prostředí je často obtížné zaručit kdo má přístup ke službám
- Zabezpečení musí být na každém rozhraní
- Kódování dat pro jednotlivé tenanty
- Oddělená správa identit
- Oddělení citlivých dat

Load ballancing

- Elastic load ballancing
- GEO load ballanacing
- Sticky session

Replikace dat

- Master-Slave
- Master-Master
- Redundance
- Automatické zotavení
- Záloha

Distribuovaná úložiště

- Například Amazon S3
- Distribované datové úložiště
- Content Delivery Network
- Přístup pomocí webových služeb
- Automatické škálování