

Dokument

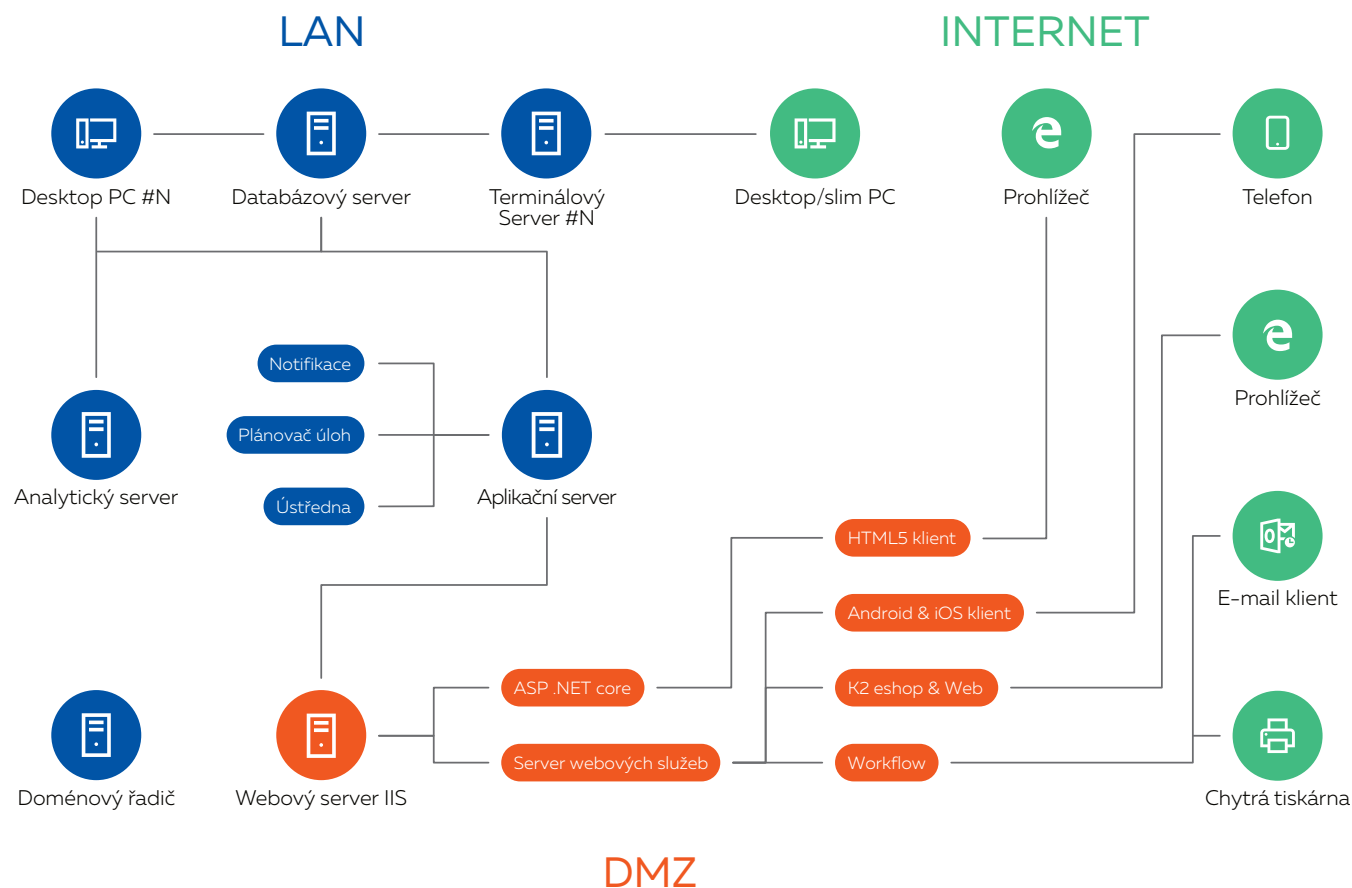
ARCHITEKTURA K2

OBSAH

1. ARCHITEKTURA K2	3
1.1. Základní popis	3
1.2. Základní architektura klient – databázový server	4
1.3. Použití terminálového serveru (popř. serverů)	4
1.4. Použití analytického serveru	4
1.5. Použití aplikačního serveru	5
1.6. Použití webového serveru	6
2. INSTALACE APLIKACÍ	7
3. KOMUNIKACE	8
3.1. Komunikace mezi internetovým prohlížečem a eShopem	8
3.2. Komunikace mezi eShopem a webovým serverem	8
3.3. Komunikace mezi aplikačním serverem a databázovým serverem	9
4. ROZLOŽENÍ KOMPONENT NA SERVERY	10
4.1. Většina služeb na jednom serveru	10
4.2. Každá služba na vyhrazeném serveru	10
4.2.1. Možné zjednodušení	11
4.2.2. Možné zvýšení odolnosti – více aplikačních a webových serverů	11
4.2.3. Doménový řadič	11

1. ARCHITEKTURA K2

1.1. ZÁKLADNÍ POPIS



System K2 používá vícevrstvou architekturu, která propojuje business logiku aplikačního serveru s databázovým serverem, terminálovým serverem, serverem pro analytické služby a webovým serverem. K těmto serverům se pak připojují klientské aplikace.

Dále budou popsány typické varianty provozu K2, které se odlišují použitými prvky. Celá logická skupina počítačů každé varianty architektury pak sdílí společnou centrální adresářovou databázi, tzn., že automatickou součástí řešení je doménový řadič, který je zodpovědný za autentizaci uživatelů v doméně Windows.

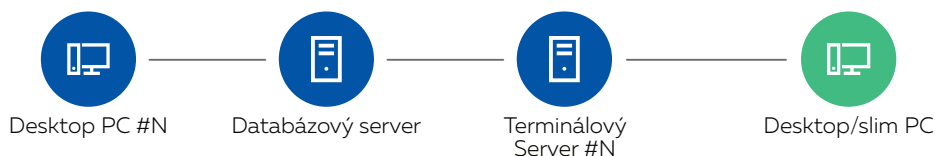
1.2. ZÁKLADNÍ ARCHITEKTURA KLIENT – DATABÁZOVÝ SERVER



Nejjednodušší variantou provozu systému K2 je spuštění desktopové aplikace K2 přímo proti databázovému serveru. Desktopová aplikace K2 pak obsahuje logiku všech funkcionalit systému K2 a pro jejich vykonání formuluje SQL dotazy, které zasílá databázovému serveru. Proto se také tato aplikace často označuje jako tzv. tlustý klient.

Jako databázový server je možno použít SQL Server firmy Microsoft a s jistými omezeními i databázový server Oracle.

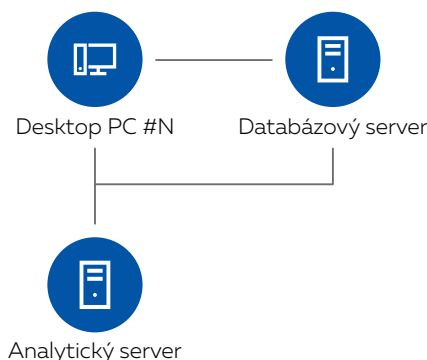
1.3. POUŽITÍ TERMINÁLOVÉHO SERVERU (POPŘ. SERVERŮ)



Zapojení terminálového serveru přináší jak možnost jednodušší správy a aktualizace systému K2 v prostředí souběžné práce stovek uživatelů, tak možnost spuštění desktopové aplikace na hardwarově levnější platformě tenkého klienta. Jde o aktuálně nejčastěji využívaný způsob provozu K2 v datovém centru.

Běžně je také využíváno několika terminálových serverů, které pak tvoří terminálovou farmu. Jako terminálový server je možno využít technologii RDP firmy Microsoft a s jistými výhradami i technologii firmy Citrix.

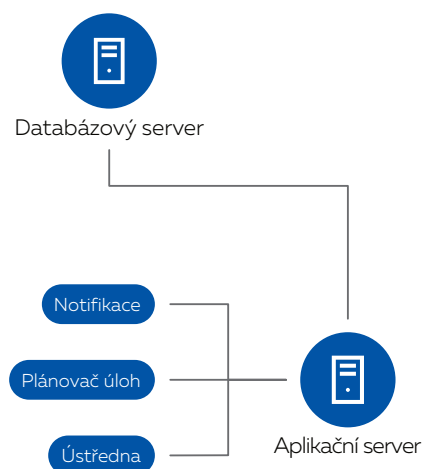
1.4. POUŽITÍ ANALYTICKÉHO SERVERU



Zapojením analytického serveru do architektury získáme v K2 možnost využívat srozumitelné obchodní analýzy v podobě dashboardů, popř. v podobě sofistikovanějšího prohlížeče kostek.

Jako analytický server je použit SQL Server Analysis Services firmy Microsoft. Export dat z K2 do kostek OLAP probíhá typicky jednou denně jako noční plánovaná úloha.

1.5. POUŽITÍ APLIKAČNÍHO SERVERU

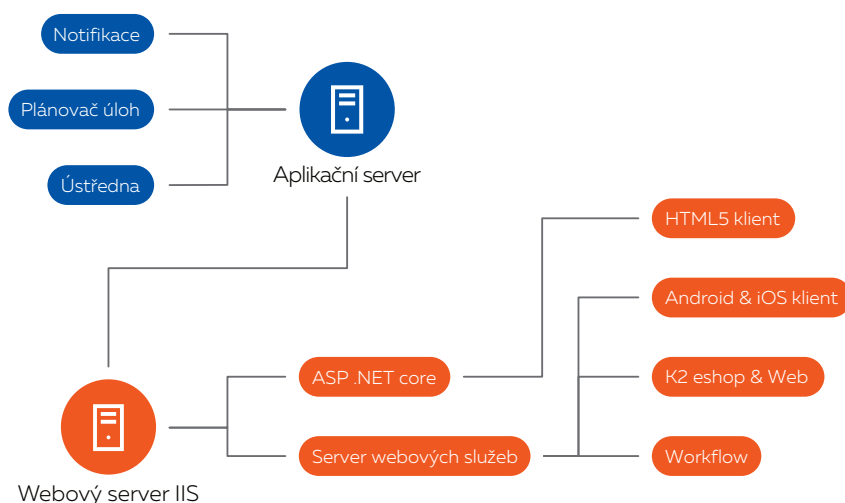


Aplikační server obsahuje podobně jako tlustý klient K2 logiku všech funkcionalit systému K2 vyjma těch, které mají přímou vizuální povahu. Je spuštěný trvale bez ohledu na to, zda je v K2 přihlášený nějaký běžný uživatel, či nikoliv. Proto se nejčastěji používá pro provoz plánovače úloh. Díky plánovači úloh lze automaticky spouštět pravidelné úlohy jako např. aktualizace kurzů, ceníků a kreditů, přepoččet eShopu a skladu nebo jednorázové nepravidelné úlohy jako např. přepoččet vybraného ceníku, vybraného zboží nebo odeslání jednorázového mailu.

Další schopností aplikačního serveru je zpracování notifikací. Díky nim je možno např. schvalovat workflow procesy přímo z mailu nebo z mobilu, aniž by bylo nutné spouštět aplikaci K2.

Další schopností aplikačního serveru je komunikace s ústřednou Asterisk. Díky ní je možno zaznamenávat telefonní hovory nebo provozovat K2 call centrum.

1.6. POUŽITÍ WEBOVÉHO SERVERU

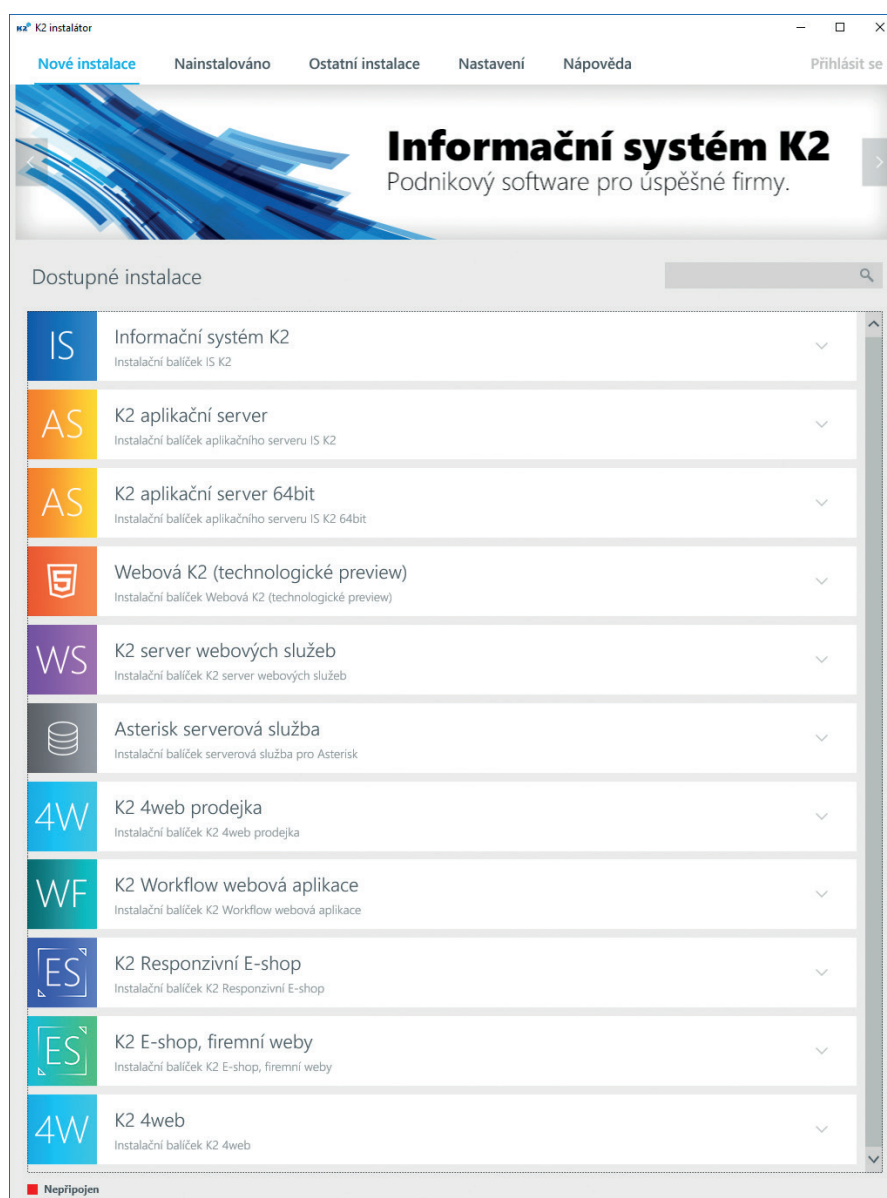


Zapojením webového serveru dochází k bezpečnému otevření systému K2 dalším stranám, jako jsou vaši zákazníci, uživatelé eShopu nebo klientským webovým aplikacím K2. Pomocí serveru webových služeb pak dokáže jakákoliv aplikace naprogramovaná v libovolném programovacím jazyce zavolat kteroukoliv funkci systému K2, kterou umí aplikační server zpracovat. Díky serveru webových služeb je možno provozovat klienty K2 na platformách Android a iOS, K2 eShop, firemní webové stránky, WMS čtečky a další aplikace.

Kromě serveru webových služeb běží na webovém serveru další služba, ASP.NET core, která slouží pro připojení HTML5 klienta K2. Díky němu lze provozovat klientskou aplikaci K2 v internetovém prohlížeči podobným způsobem a s podobným vzhledem, jak jsou uživatelé zvyklí z desktopové aplikace K2.

Protože je pro běh této aplikace dostačující pouze internetový prohlížeč, bývá označována jako tenký klient.

2. INSTALACE APLIKACÍ

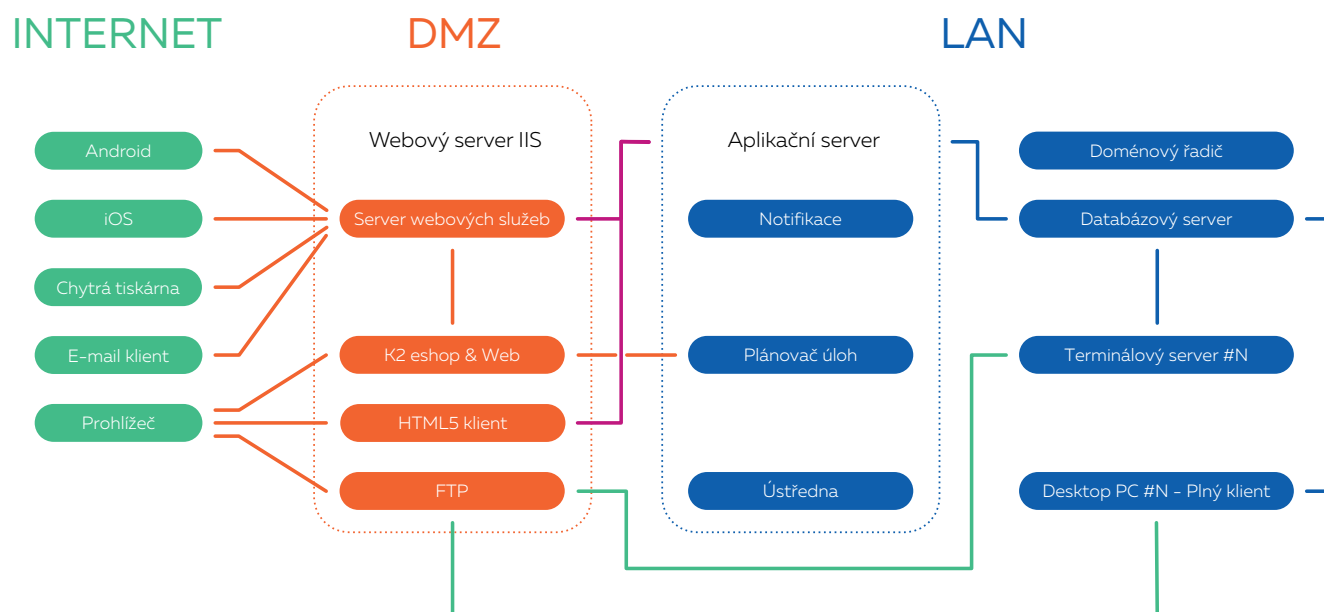


K instalaci klientských i serverových aplikací slouží K2 instalátor. Hlavní možností instalátoru je produkt „Informační systém K2“, který nainstaluje serverovou část systému a desktopového klienta K2. Další možností je aplikační server a to jak ve verzi 32b, tak ve verzi 64b. Pro komunikaci s ústřednou asterisk slouží „Asterisk webová služba“.

Pro využití webové K2 (HTML5 klienta), eShopu, firemních webů je nutná instalace nejen příslušné aplikace, ale také instalace webových služeb „K2 server webových služeb“. Ty jsou navíc použitelné i samostatně pro aplikace třetích stran.

Podrobný popis postupu instalace jednotlivých produktů je uveden v dokumentaci K2 v dokumentu K2_Instalátor_K2_(verze K2).pdf.

3. KOMUNIKACE



Jednotlivé komponenty architektury mezi sebou komunikují pomocí komunikačních protokolů a portů. Následuje jejich souhrn a popis principu komunikace.

— Protokol: http(s), port: 443, 80

— Protokol: Named pipe, port: 445

— Protokol: MS SQL Server: Named pipe, TCP, port: 445, 1433

Analysis services: TCP port: 2383

— Protokol: FTP(S), port: 21, 990

3.1. KOMUNIKACE MEZI INTERNETOVÝM PROHLÍŽEČEM A ESHOPEM

Uživatel kontaktuje eShop pomocí URL adresy (např. <http://go.k2.cz/eshop>). Komunikace probíhá prostřednictvím HTTP nebo HTTPS protokolu. Na portu, který je definován na webovém serveru (IIS).

3.2. KOMUNIKACE MEZI ESHOPEM A WEBOVÝM SERVEREM

Aplikace při renderování výsledné HTML stránky (jako výsledek prohlížeči) načítá data prostřednictvím K2 serveru webových služeb. S tím komunikuje pomocí HTTP(S) protokolu. eShop tedy zasílá dotazy pomocí metod GET, POST, PUT a výsledná data získává v JSON formátu.

3.3. KOMUNIKACE MEZI APLIKAČNÍM SERVEREM A DATABÁZOVÝM SERVEREM

Aplikační server zpracovává jednotlivé požadavky ze serveru webových služeb, případně dalších klientů připojených ve vnitřní síti. Každý požadavek je při svém vstupu zařazen do nejméně vytížené fronty (počet front je závislý na počtu sdílených uživatelů v licenci K2). Fronty jsou paralelně zpracovávány aplikačním serverem, v rámci jedné fronty jsou požadavky serializovány.

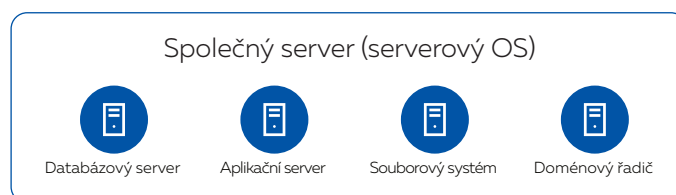
Aplikační server data získává z:

1. Databázového serveru
 - › Dle nastavení v k2.ini prostřednictvím: Named pipe, port 445 nebo TCP port: 1433
2. Analytického serveru
 - › TCP port: 2383
3. Souborového systému
 - › Nutnost viditelnosti složek systému K2

4. ROZLOŽENÍ KOMPONENT NA SERVERY

Konkrétní rozložení prvků architektury na jednotlivé logické servery a nastavení omezení využití hardwarových komponent je předmětem projektu implementace, který musí vzít v úvahu běžné i maximální intenzity zátěže jednotlivých prvků architektury vůči provozovanému hardware. Dále budou uvedena některá typická rozložení jednotlivých komponent architektury na logické servery.

4.1. VĚTŠINA SLUŽEB NA JEDNOM SERVERU



Tato varianta uspořádání je velmi úsporná jak z hlediska počtu serverů a jejich operačních systémů, tak i z hlediska následné správy. Vzhledem k nízké odolnosti vůči přetížení je doporučena pro testovací, vývojové a laboratorní účely.

Typickou komplikací tohoto řešení je spotřeba paměti databázového serveru v default nastavení. Proto je nutné zajistit dostatek paměti pro jednotlivé služby (minimálně např. 3–4GB na každou službu). Další komplikací je pak vytížení CPU, které lze omezit jen neefektivně pomocí afinity.

Výhody:

- › Menší počet serverů
- › Cena

Nevýhody:

- › Nutnost vyhrazení hardwarových prostředků pro jednotlivé služby
- › Vysoké nebezpečí kolapsu všech služeb systému najednou
- › Nedoporučeno firmou Microsoft

4.2. KAŽDÁ SLUŽBA NA VYHRAZENÉM SERVERU

Tato varianta je doporučena jako provozní z důvodu škálování hardwarových prostředků na jednotlivé služby přesně podle jejich předpokládaného využití, a především z důvodu bezpečnějšího provozu, který je odolný vůči přetížení jednotlivé služby.

Výhody:

- › Škálování hardwaru
- › Odolnost proti výpadku jednotlivých služeb

Nevýhody:

- › Větší počet serverů
- › Cena

4.2.1. MOŽNÉ ZJEDNODUŠENÍ



Variantu lze částečně zjednodušit tak, že na databázový server umístíme i souborový systém K2, protože databázový server dokáže hospodařit s pamětí operačního systému inteligentně (dokáže ji po přidělení i vrátit – s jistou časovou prodlevou), navíc intenzita práce standardu K2 se souborovým systémem je velmi nízká.

4.2.2. MOŽNÉ ZVÝŠENÍ ODOLNOSTI – VÍCE APLIKAČNÍCH A WEBOVÝCH SERVERŮ

Variantu lze rozšířit v případě využití více aplikačních či webových serverů, např. pokud je určitému dodavateli vyhrazeno konkrétní API na dané URL adrese, je vhodné pokud příslušný aplikační server pracuje na samostatném logickém serveru. Stejně tak je možné použít více aplikačních serverů na samostatných serverech pro služby plánovače úloh, notifikací nebo komunikace s ústřednou.

4.2.3. DOMÉNOVÝ ŘADIČ

Z důvodu bezpečnosti je doporučeno provozovat doménový řadič na vyhrazeném serveru.