



IBM IDC Brno

# SSO & DCS II

Eva Soldánová



# Obsah

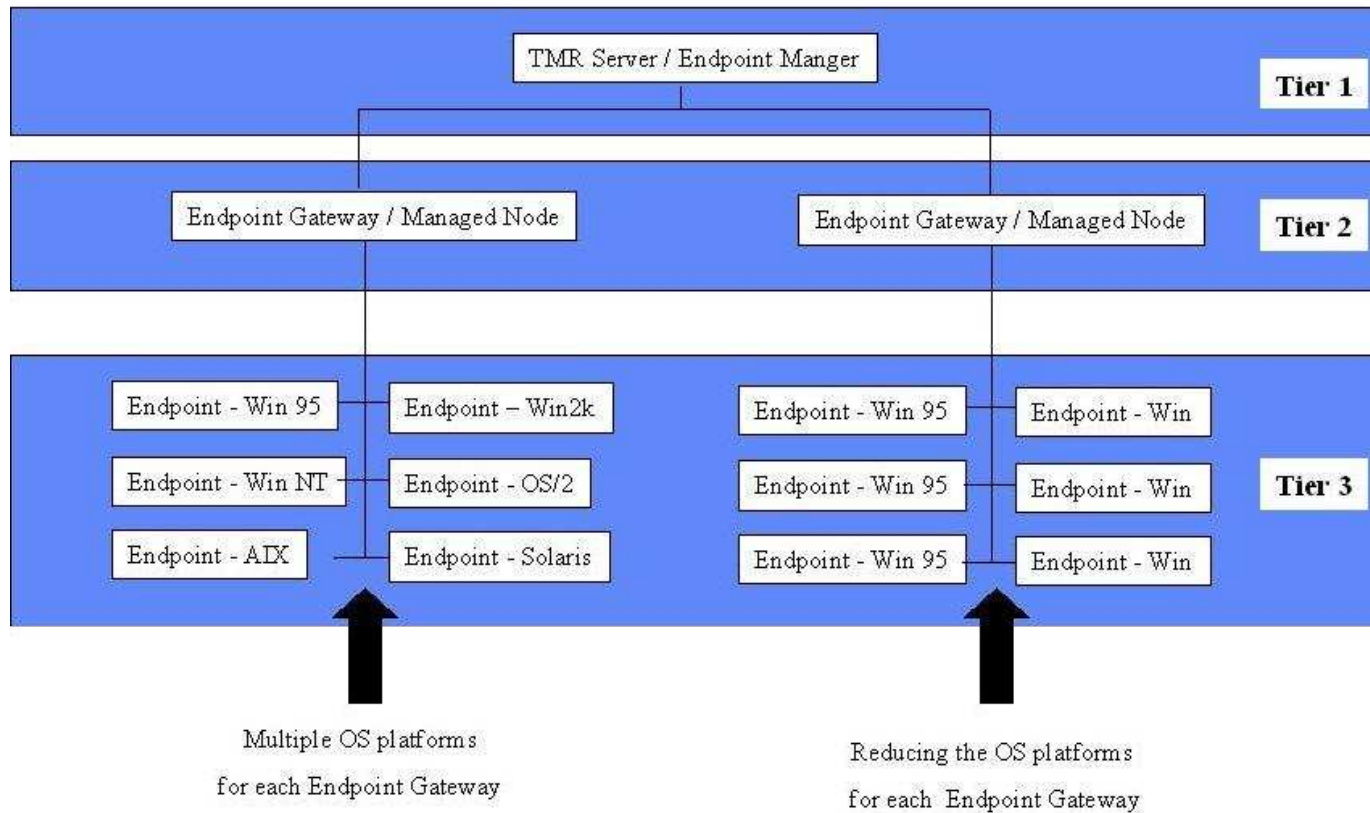
- **Infrastruktura**
- **Architektura**
- **TMR**
- **OS and SW**
- **Procesy**
- **ITM**
- **Heartbeat**
- **Protokoly**
- **Proaktivní monitoring**
- **Root-cause analysis**
- **Monitory**
- **Příklad vygenerování alertu**
  
- **Příkazy**
- **Zkratky**

## Tivoli infrastruktura

- **Tivoli je rodina produktů používaná pro správu počítačových systémů.**
- **Jádro Tivoli produktů tvoří:**
  - Tivoli Framework = základ pro všechny ostatní Tivoli produkty
  - Tivoli Monitoring
  - Tivoli Storage Manager (TSM) – Tivoli backup
  - Tivoli Software Distribution (TSD) – instalace SW

# Tivoli architektura

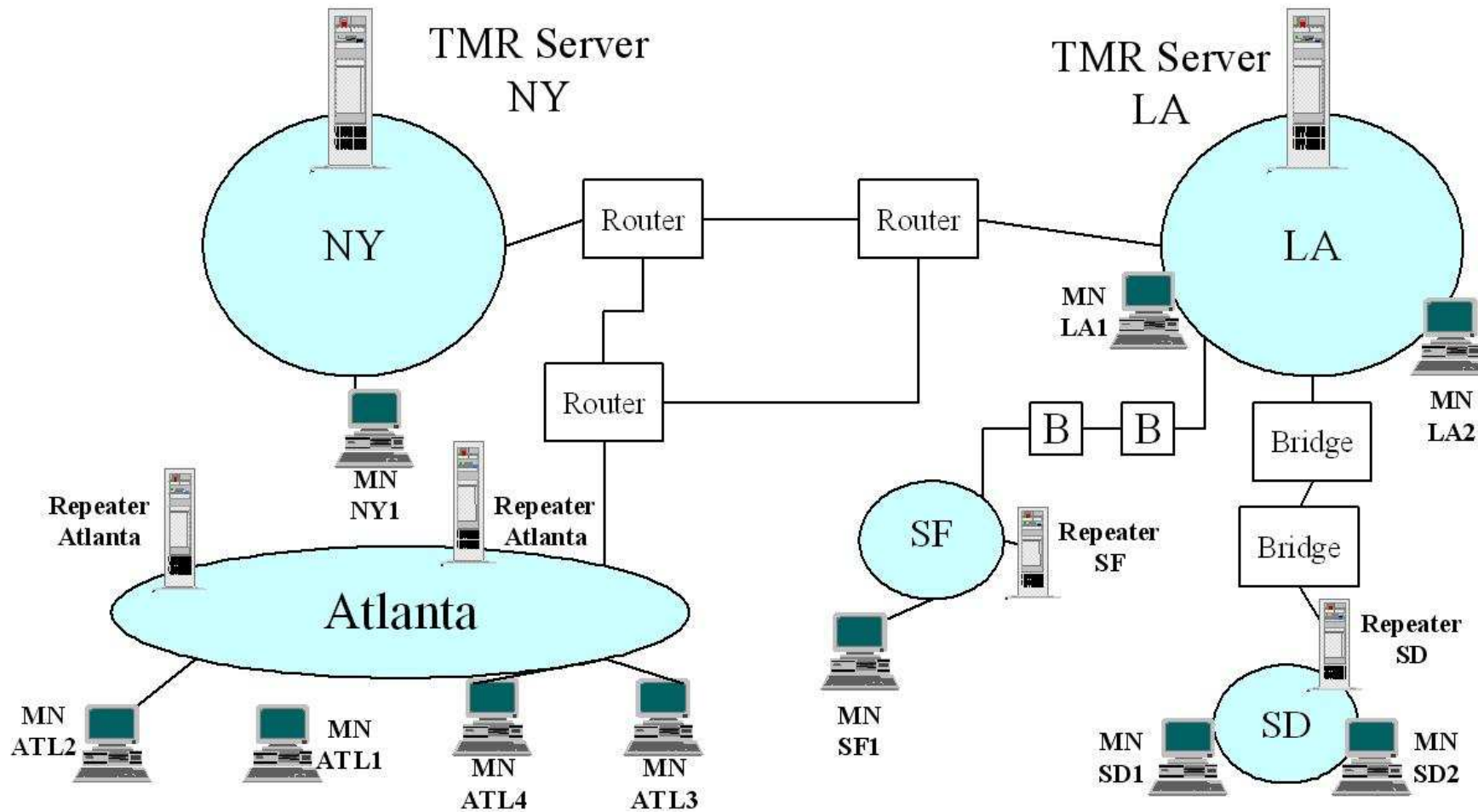
## 3-tier Architecture



## Tivoli Management Region (TMR)

- Každá oblast (TMR) je tvořena jedním TMR serverem, jednou nebo více *Gatewayemi* a koncovými uzly
- TMR server je centrální bod správy celé oblasti (TMR), jež umožňuje provádět administraci pro danou oblast.
- Gatewaye jsou používány ke zvyšování výkonu TMR a rozdělení TMR na logické a fyzické části.
- poznámka: Na jednom počítači může být TMR server, gateway i koncový uzel.

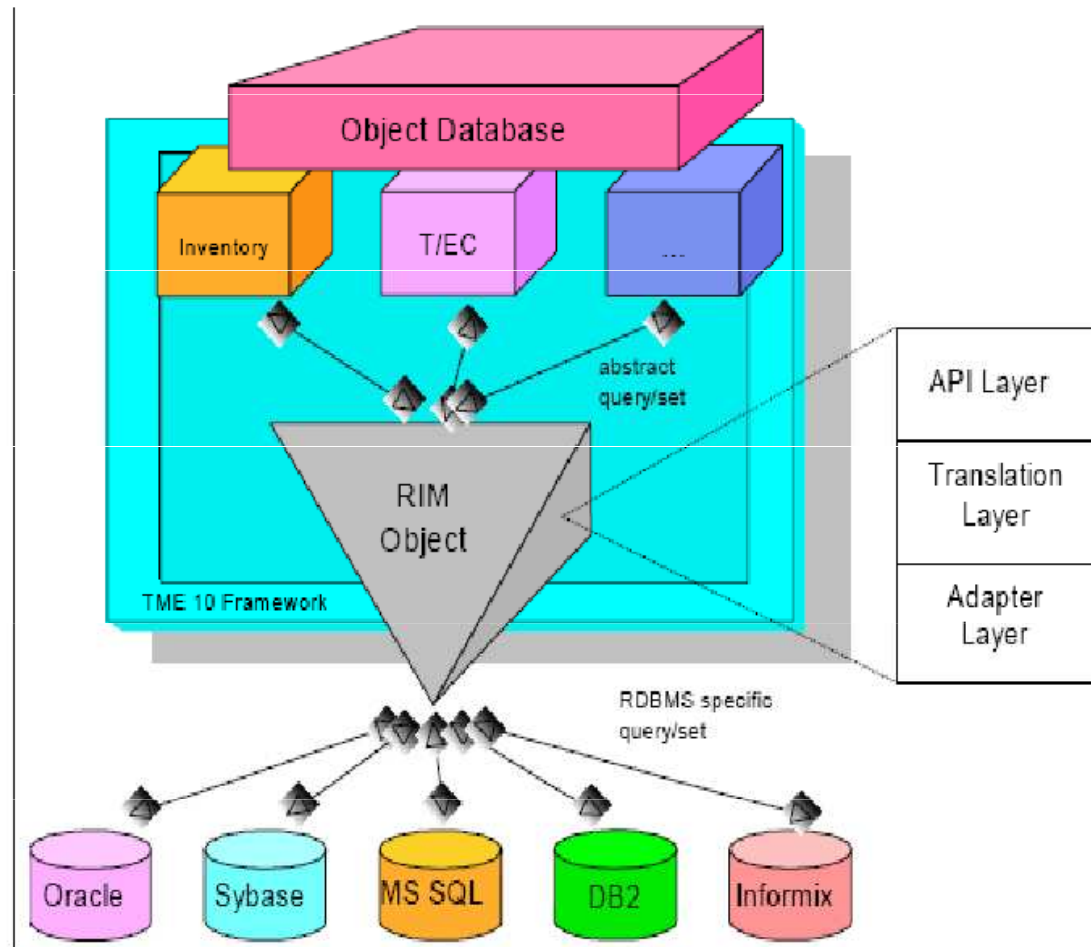
# TMR



## Kde co

- **Server – IBM AIX, Sun Solaris, HP-UX, MS Windows, Red Hat, SuSE, Turbolinux**
- **Gateway – všechny výše zmíněné + Novell Netware, IBM OS/2**
- **Koncové uzly – všechny výše zmíněné + IBM OS/400, Nokia Communicator, PalmOS, PocketPC**
- **RDBMS – DB2, MSSQL, Oracle, Sysbase, Informix**

# Relational DataBase Management System (RDBMS)





# Procesy

## ■ **oserv**

- Hlavním Tivoli procesem je oserv, který musí běžet na TMR serveru a gatewayích.
- Oserv řídí rozsáhlou objektovou databázi, která obsahuje všechny objekty v TMR (koncové uzly, gatewaye, ...)

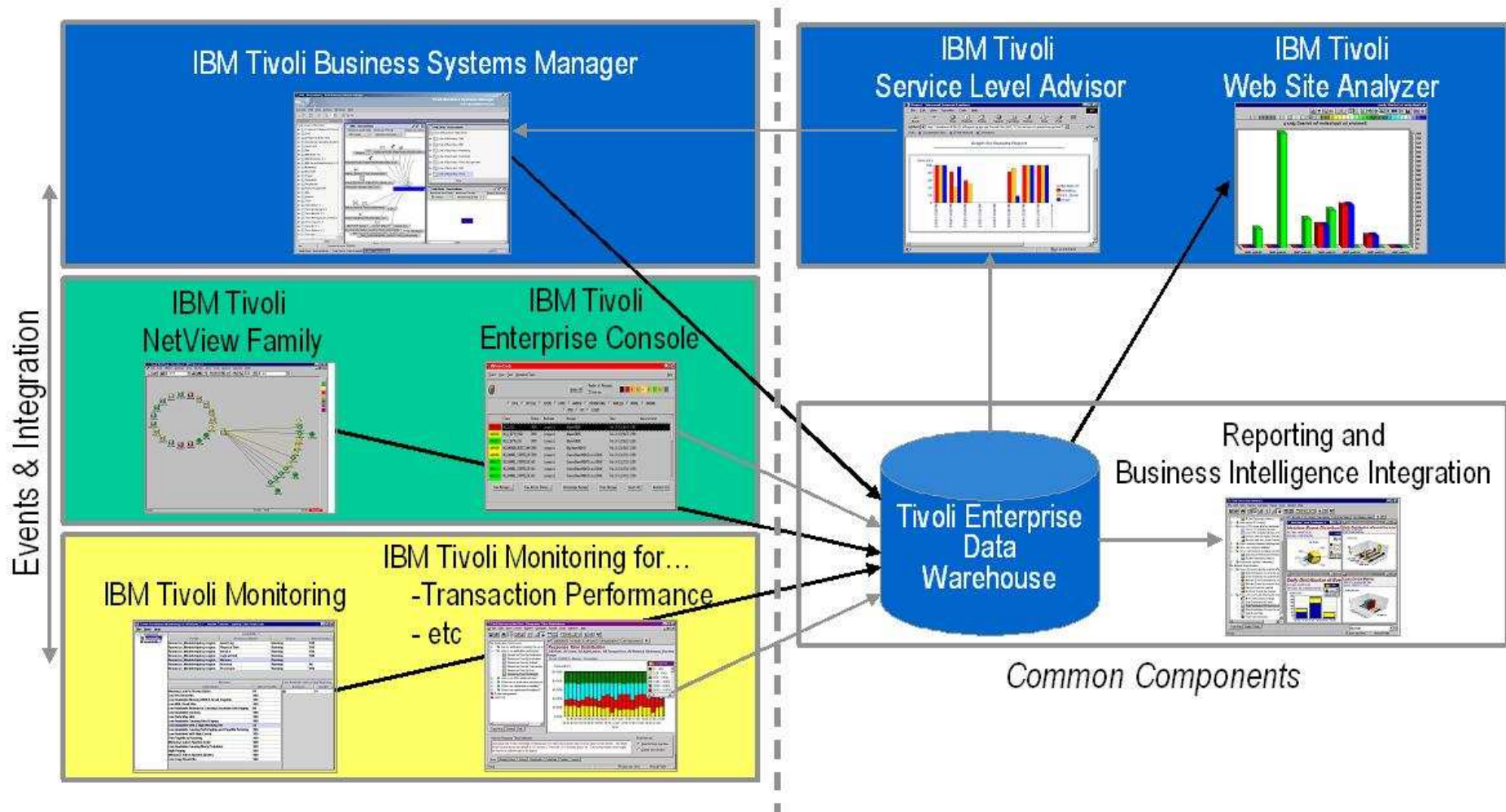
## ■ **lcfcd**

- Na koncových uzlech musí běžet proces lcfcd = Lightweight Client Framework Daemon.

# ITM

## Real-time Operations

## Predictive Operations



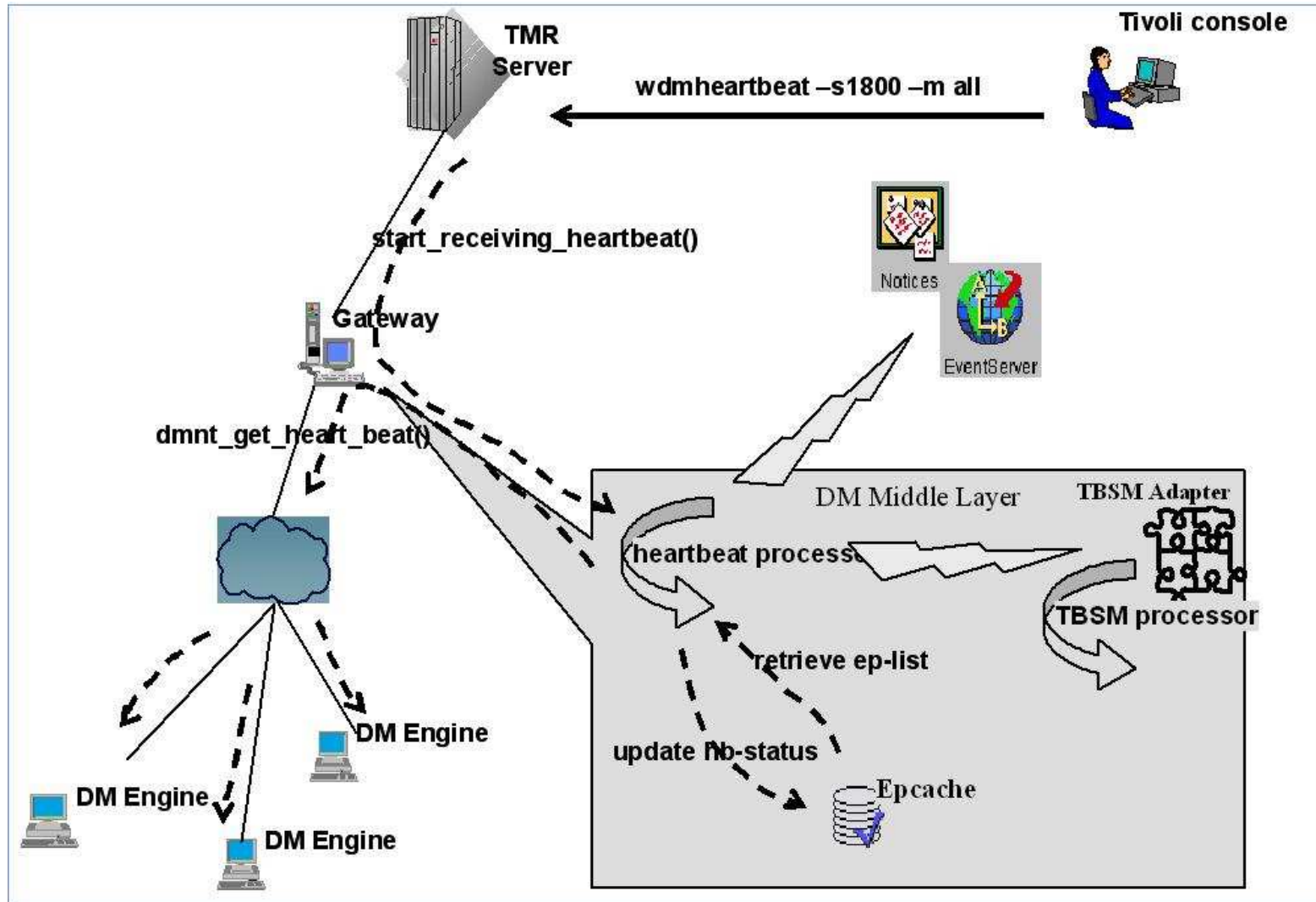
## Základní pojetí monitoringu I

- **IBM Tivoli Monitoring (ITM) zajišťuje monitoring důležitých systémových zdrojů, detekuje překážky a potencionální problémy a automaticky pomáhá z kritických situací.**
- **ITM oprošťuje systémové administrátory od manuálního prohlížení rozsáhlých systémů a pomáhá řešit potencionální či aktuální problémy.**

## ITM na Gatewayi

- **Shromažďuje data ze všech koncových uzlů a následně je přeposílá na TEC server nebo TBSM server.**
- **Na Gatewayi musí být následující komponenty:**
  - Tzv. Task Engine – dostává požadavky a vykonává odpovídající procesy
  - **Heartbeat Processor** – monitoruje stav koncových uzlů dané gatewaye, dává nám jistotu, že Tivoli monitorovací infrastruktura běží
  - Kolektor – využívá se pro sběr monitorovaných dat
  - TBSM adaptér – dovoluje adresovat události generované ITM k TBSM rozhraní
  - TEC gateway – slouží k bezpečnějšímu posílání události v ITM

# Tok dat



# Heartbeat

- **Monitorovací stroj je zapsán na gateway při spouštění nebo když:**
  - jsou data aktualizována v cache paměti - gateway přijímá zprávu z koncového uzlu říkající, že koncový uzel byl nastartován
  - Tzv. model zdrojů je poprvé umístěn na koncový uzel
  - V koncovém uzlu je restartován Tivoli monitorovací stroj
- **Heartbeat procesor pravidelně monitoruje koncové uzly, pak gateway obdrží status z monitorovacího stroje v nastavitelných intervalech**
- **Heartbeat procesor může zaznamenávat status koncového uzlu ve vlastní cache paměti, která je rozdělena do dvou skupin**
  - informační            - žije/heartbeat byl stopnut
  - chybová              - Tivoli monitorovací stroj byl stopnut
    - koncový uzel není v síti dostupný
    - tzv. model zdrojů je v chybovém stavu



# Protokoly I

- **TCP/IP - internetové síťové protokoly, např:**
  - Aplikační protokoly (FTP, Telnet, HTTP, SSL, IMAP, DNS, NFS, ...)
  - Všechny OS podporují protokol TCP/IP obsahující program **ping**, kterým uživatel může na cílový uzel odeslat žádost o **echo**. Program ping pak zobrazuje odpověď.

```
D:\>ping 194.149.105.18
```

```
Pinging 194.149.105.18 with 32 bytes of data:
```

```
Reply from 194.149.105.18: bytes=32 time<10ms TTL=63  
Reply from 194.149.105.18: bytes=32 time<10ms TTL=63  
Reply from 194.149.105.18: bytes=32 time<10ms TTL=63  
Reply from 194.149.105.18: bytes=32 time<10ms TTL=63
```

- Systém odeslal čtyřikrát žádost o echo. Odpověď měla 32 bajtů dlouhou část a získal ji do 10 ms.

## Protokoly II

### ■ ICMP

- formálně se jedná o součást protokolu IP, ale chová se jako protokol vyšší vrstvy
- mechanismus pro ošetření zahlcení linek
- Zahrnuje jednoduchý nástroj **Echo** = žadatel vysílá ICMP-paket „žádost o echo“ a cílový uzel je povinen odpovědět ICMP-paketem „Echo“.
- slouží zejména k signalizaci mimořádných stavů v síti postavených na IP-protokolu, to může být:
  - nedosažitelná síť, uzel, protokol, port,
  - explicitní směrování selhalo,
  - adresátova síť je neznámá,
  - adresátův uzel je neznámý,
  - čas vypršel,
  - ...

### ■ SNMP – slouží ke správě sítí, směrování paketů



## Proaktivní monitoring

- **Rychlé detekování a reagování na možné problémy**
- **Automatické rozpoznání stavu, který hrozí při překročení stanovených hraničních hodnot**
- **Automatické provádění opravných akcí k ozdravení systému**
- **Rozpoznání problému před dopadem na koncového uživatele**

## Root-cause analysis (RCA)

- **Tento nástroj jako součást ITM pomáhá určit kritické situace.**
  - Normy – standardy definované vztahy mezi obchodními objekty a událostmi spojené se slabými výkony nebo výpadky.
  - Logika rozhodovacích stromů – Tivoli SW používá tuto logiku k aplikování několika pravidel, aby ověřil míru zdraví systému a rozhodl o případných nápravných akcích.
  - Intelligence – ITM může zbavit systémové administrátory nudných úkolů tím, že poskytuje cenné informace pro troubleshooting kritických situací.

# Monitory I

- **Na každém monitorovaném zařízení je rozmístěno několik Tivoli monitorů. Každý je pak odpovědný za monitoring specifických zdrojů, např.:**
  - **Disk monitor** monitoruje spotřebované místo na disku
  - **Proces monitor** monitoruje, zda monitorované procesy běží
  - **Oracle monitor** dohlíží na Oraclovské instance
- **Nejvíce alertů je tedy generováno přímo z monitorovaných uzlů, ale existují i další typy alertů, např.:**
  - zda je server dostupný hlídá **Tivoli Netview** – který jen ověřuje zda jsou síťová rozhraní dosažitelná
  - Tzv. URL alerty jsou generovány tzv. **PCPMM**, který pomocí šablony ověřuje, zda jsou příslušné stránky dostupné. Jedná se o tzv. UP/DOWN monitoring, který je prováděn na specializovaném serveru.

## Monitory II

- Jeden problém může generovat alerty z různých monitorů

SLD_icsdbf	2006-2-1 19:32:42	ORACLELGE104E: PROD Database KO - Status :UNKNOWN Monitoring Failed	Top_Oracle_monitor
SLD_icsdbf	2006-2-1 19:31:43	GENPROLGE002I Daemon: ora_smon_PROD down ( Impacts oracle)	Generic_monitor_proces
SLD_icsdbf	2006-2-1 19:31:42	GENPROLGE002I Daemon: ora_reco_PROD down ( Impacts oracle)	Generic_monitor_proces
SLD_icsdbf	2006-2-1 19:31:42	GENPROLGE002I Daemon: ora_pmon_PROD down ( Impacts oracle)	Generic_monitor_proces
SLD_icsdbf	2006-2-1 19:31:42	GENPROLGE002I Daemon: ora_lgwr_PROD down ( Impacts oracle)	Generic_monitor_proces
SLD_icsdbf	2006-2-1 19:31:42	GENPROLGE002I Daemon: ora_dbw0_PROD down ( Impacts oracle)	Generic_monitor_proces
SLD_icsdbf	2006-2-1 19:31:41	GENPROLGE002I Daemon: ora_ckpt_PROD down ( Impacts oracle)	Generic_monitor_proces
SLD_icsdbf	2006-2-1 19:31:41	GENPROLGE002I Daemon: ora_arc0_PROD down ( Impacts oracle)	Generic_monitor_proces

## Příklad vygenerování alertu

- Rozsah diskového místa na nějakém filesystemu překročil definovanou hranici.
- Disk monitor zjistí tuto situaci.
- Následně je vygenerována Tivoli událost, která se pošle na TEC server.
- TEC server ověří událost a pošle ji na konzoli.

## Užitečná příkazy

- **wep <endpoint\_name> status ...** ověří koncový uzel (status Isfd)
- **wep <endpoint\_name> ...** vrátí detaily o monitorovaném uzlu
- **wep ls ...** seznam všech monitorovaných uzlů
- **wlseng -z <endpoint\_name> ...** vypíše detaily o profilech rozmístěných na monitorovaném uzlu, nahodí dm\_ep\_engine
- **wstopeng -z <endpoint\_name> ...** zastaví dm\_ep\_engine na monitorovaném uzlu
- **report -ep <endpoint\_name> ...** vrátí stav monitorů rozmístěných na monitorovaném uzlu
- **wping <gateway\_name> ...** ověří status oserv procesu na gatewayi
- **wgateway ...** seznam gatewayi a jejich status na TMR
- **wgateway <gateway\_name> restart ...** restartuje oserv na gatewayi
- **odadmin odlist ...** vypíše všechny gatewaye na TMR s detaily o Tivoli ID (číslo objektu, číslo dispečeru)

Poznámka: všechny výše zmíněné příkazy je nutné provádět na TMR

## Použité zkratky

- **Dm\_ep\_engine = Distributed Monitoring EndPoint Engine**
- **ICMP = Internet Control Message Protocol**
- **ITM = IBM Tivoli Monitoring**
- **Icfd = Lightweight Client Framework Daemon**
- **mn = managed node**
- **OS = operation systém**
- **RCA = Root-Cause Analysis**
- **RDBMS = Relation database management systém**
- **SNMP = Simple Network Management Protocol**
- **TBSM = Tivoli Business Systems Manager**
- **TCP/IP = Transmission Control Protocol/Internet Protocol**
- **TEC = Tivoli Enterprise Console**
- **TSD = Tivoli Software Distribution**
- **TSM = Tivoli Storage Management**
- **TMR = Tivoli Management Region**
- **SW = software**



# Odkazy

- <http://publib.boulder.ibm.com/tividd/td/tdprodlist.html>
- <http://www.redbooks.ibm.com/redbooks/pdfs/sg245240.pdf>