

# Nezávislé komponenty Komunikace komponent

Tomáš Černý

# Zlepšení komunikace

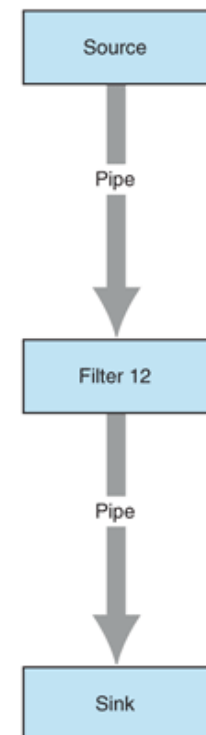
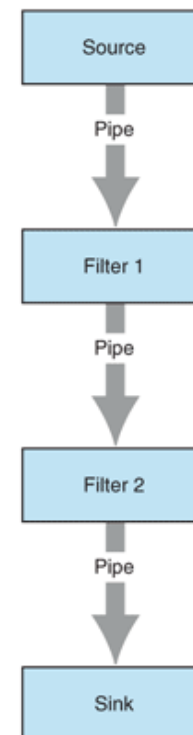
- Nezávislé komponenty
- Komunikace mezi procesy = předávání zpráv
- K naplnění
  - modifikovatelnosti
  - škálovatelnosti
  - výkonu
  - a konfigurace
- Mnoho variant, které naplňují různé cíle

# Varianty

1. Jednocestný datový tok – síť filtrů
2. Klient-server
3. Heartbeat
4. Sonda echo
5. Token passing
6. Replikace serverů
7. Decentralizace

# Jednocestný datový tok - síť filtrů

- Datový tok přes komunikující procesy
  - abstrakce data-flow (pipe-n-filter)
- Data prochází inkrementálně transformacemi
  - Transformace je samostatný proces
- Př. Strom procesů
  - formujících a třídících sítí
- Tok dat a řízení
  - jde jedním směrem

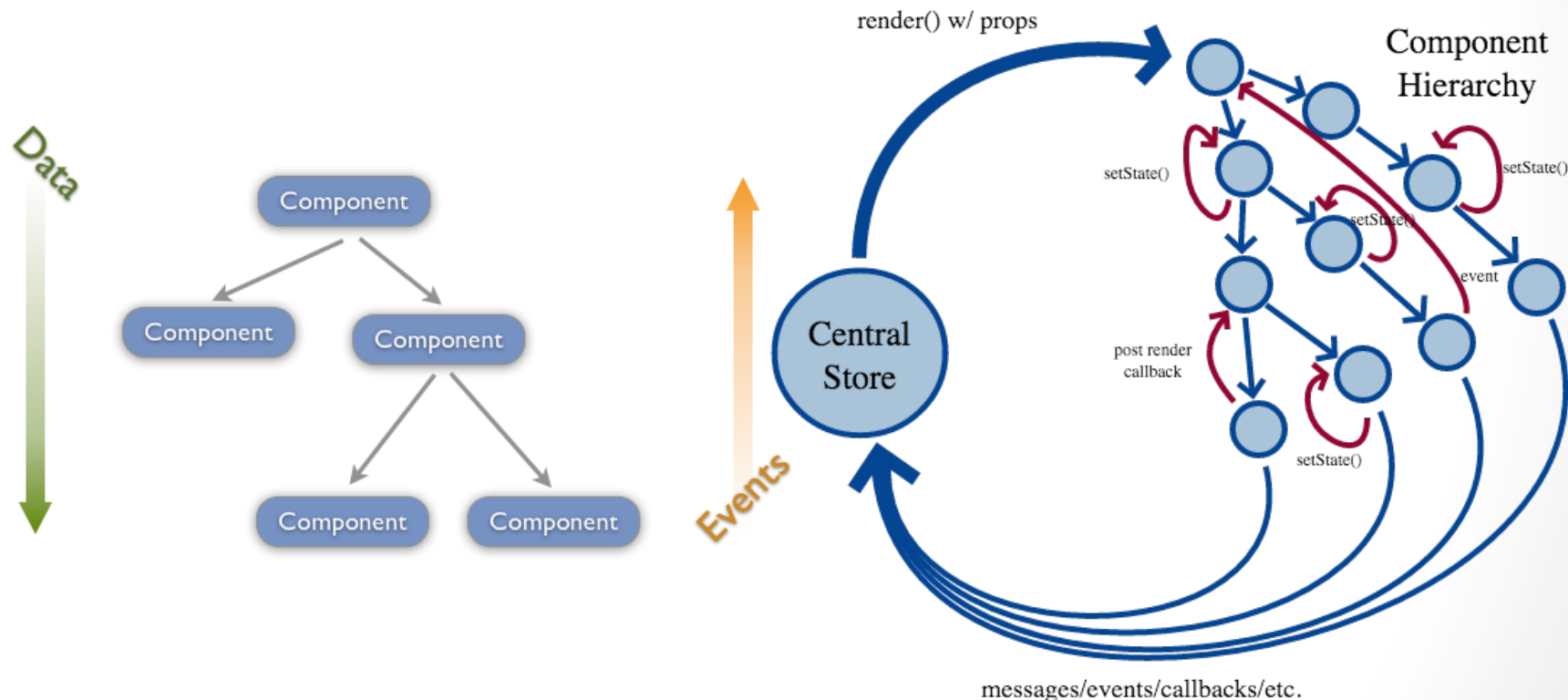


# Facebook React : one-way data flow

- React is all about **one-way data flow** down the component hierarchy. It may not be immediately clear which component should own what state.
1. Identify every component that renders something based on that state.
  2. Find a common owner component (a single component above all the components that need the state in the hierarchy).
  3. Either the common owner or another component higher up in the hierarchy should own the state.
  4. If you can't find a component where it makes sense to own the state, create a new component simply for holding the state and add it somewhere in the hierarchy above the common owner component.

# Facebook React : one-way data flow

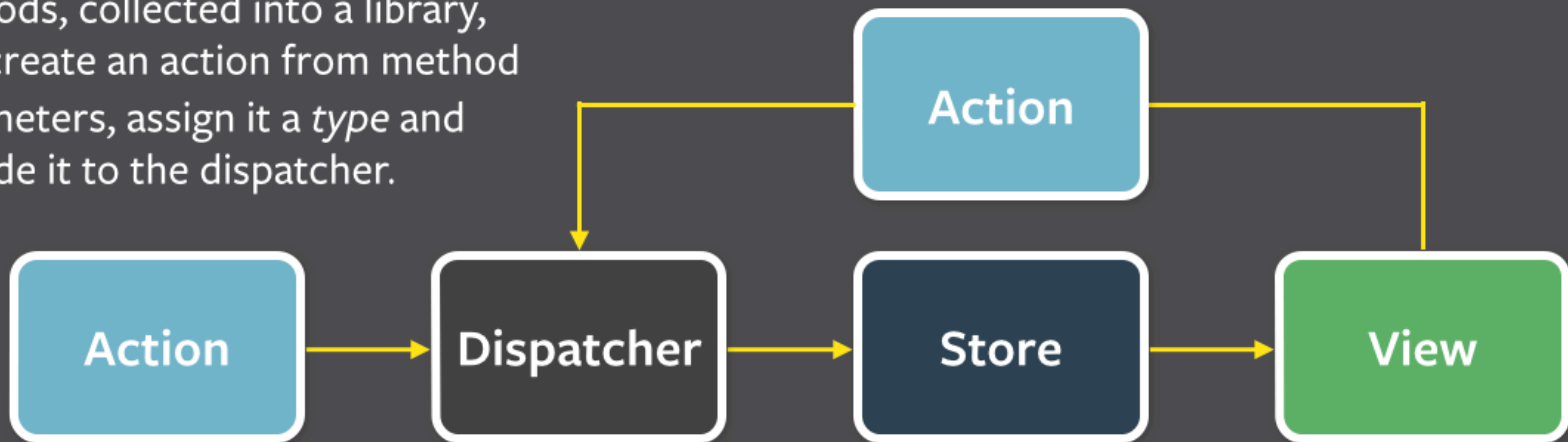
- React is all about **one-way data flow** down the component hierarchy. It may not be immediately clear which component should own what state.



# Facebook React : one-way data flow

## ■ Data Architecture = Response to MVC

*Action creators* are helper methods, collected into a library, that create an action from method parameters, assign it a *type* and provide it to the dispatcher.



Every action is sent to all stores via the *callbacks* the stores register with the dispatcher.

After stores update themselves in response to an action, they emit a *change* event.

Special views called *controller-views*, listen for *change* events, retrieve the new data from the stores and provide the new data to the entire tree of their child views.

# Požadavek a odpověď : Klient - Server

- Specializace komunikujících procesů
- Uvažujeme beze stavu pro interakce
  - Řízení a data jsou opačným směrem





# Požadavek a odpověď : Klient - Server

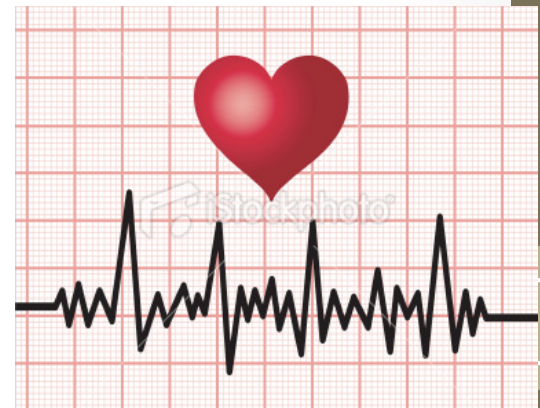
- Většina web-aplikací
- The client–server model of computing is a distributed application structure
- Partitions tasks or workloads between the *providers of a resource or service*, called **servers**, and *service requesters*, called **clients**.

Heartbeat = tam a zpět : interakce sousedů

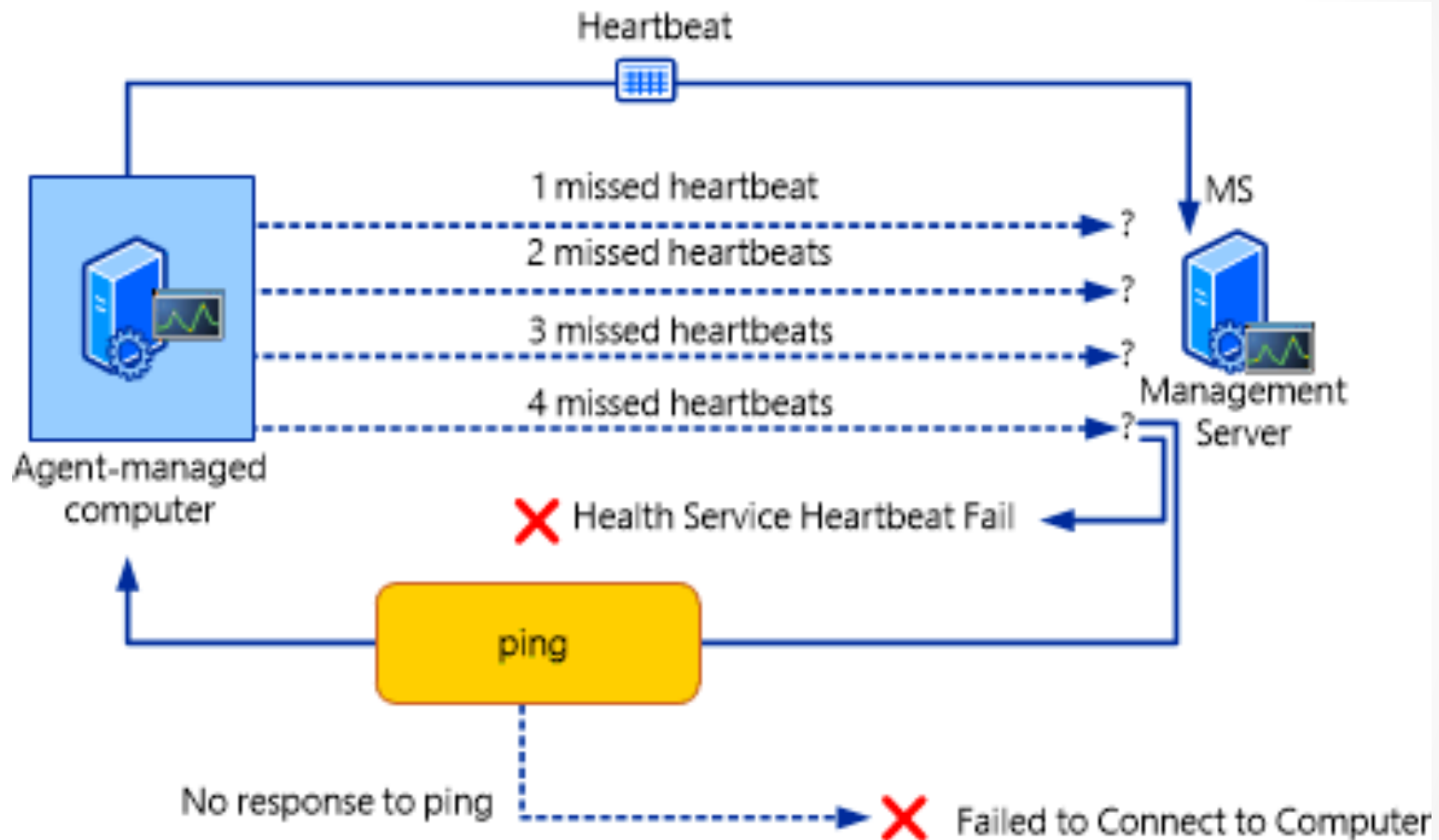
In computer clusters, heartbeat network is a private network, which is shared only by the cluster nodes, and is not accessible from outside the cluster. It is used by cluster nodes in order to monitor each node's status and communicate with each other.

### **Alias:**

- Gossip protocol
- Ping-Pong



# Heartbeat = MS System Center 2012



# Heartbeat = MS System Center 2012

Operations Manager uses heartbeats to **monitor communication channels** between an agent and the agent's primary management server.

A heartbeat **is a packet of data sent** from the agent to the management server on a regular basis, by default **every 60 seconds**, using port 5723 (UDP).

**When an agent fails to send a heartbeat 4 times**, a Health Service Heartbeat Failure **alert is generated** and **the management server attempts** to contact the computer by using **ping**. If the computer does **not respond** to the ping, a Failed to Connect to Computer **alert** is generated.

# Heartbeat = tam a zpět : interakce sousedů

- Každý uzel v procesním grafu
  - Posílá informace
  - Sbírá nové informace
  - Uzel si udržuje matici sousedů
- *Př. Prohledávání topologie sítě*
- *V každé iteraci*
  1. Na každý „tep“ každý uzel pošle sousedům svou matici
  2. Mezi heartbeaty kalkuluje matice od sousedů a upravuje svou
  3. Po „n“ iteracích zná topologii vzdálenosti „n“
  4. Konec při naplnění podmínek
    - průměr sítě, striktní n, neměnné hodnoty

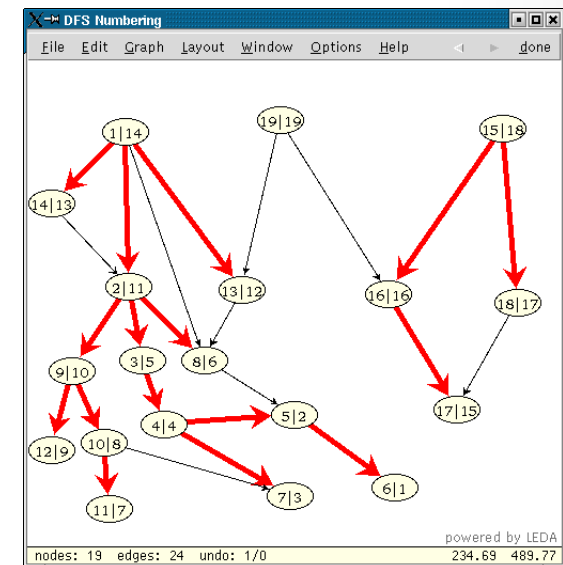
# REDIS

<http://cristian.regolo.cc/2015/09/05/life-in-a-redis-cluster.html>

- Automatic data sharding across multiple nodes
- High degree of fault tolerance
- Nodes in a Redis Cluster talk to each other using the Cluster Bus
- Nodes of the cluster exchange PING & PONG packets over the bus
- This packet flow constitutes the heartbeat mechanism of the cluster
- Each ping/pong packet carries important pieces of information about the state of the cluster from the point of view of the sender node
  - <https://youtu.be/0BLmM73y3oo>
- Redis Cluster uses a **Gossip protocol** in order to **quickly spread information through any connected node**.
- Heartbeat packets carry information on their own, but they also contain a special header for gossip data.

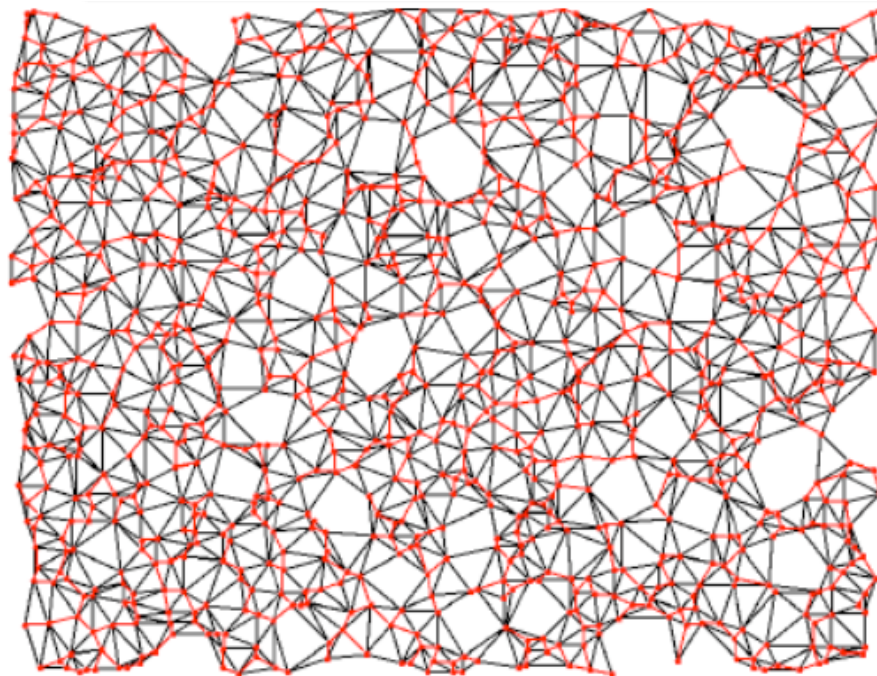
# Sonda a echo

- Sběr či distribuce dat v grafu či stromu
- Žádné iterace
  - Jeden uzel sbírá topologii od všech ostatních
  - Iniciace broadcast sonda sousedům
  - Přeposlání sondy sousedy
  - Echo jako odpověď sondám
  - Závěrem broadcast všem informace o topologii
- Paralelní zpracování analogické pro
  - Depth-First Search
- Příklady
  - Objevení topologie sítě
  - Broadcast skrze sousedy



# Sonda a echo

- Bez sdílení informací jen zprávy – příjem / odeslání
- Nodes send **probes to their neighbours** and **receive echos** from them.





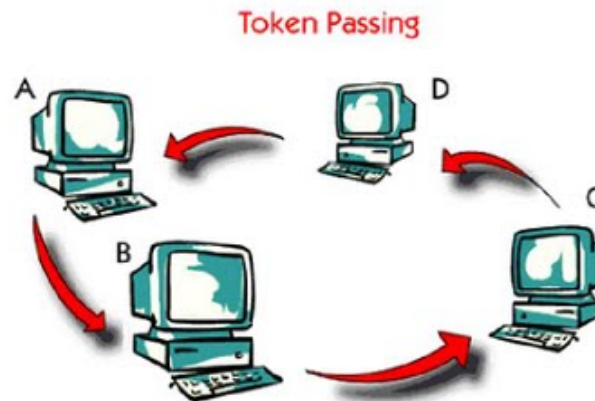
# Heartbeat vs Sonda a echo

- Heartbeat
  - Symetrický alg.
  - Mnoho zpráv
  - Paralelní iterace
- Probe-echo
  - Asymetrický
  - Méně zpráv
  - Více efektivní

# Token passing

- Token – pešek
  - Message passing
    - Reliable vs Unreliable
    - Blocking vs Non-blocking
- Distribuovaná koordinace
  - Kdo má peška má i práva
  - Sběr stavových informací
- Př. Globální stav distrib
- Př. Distribuované zámky

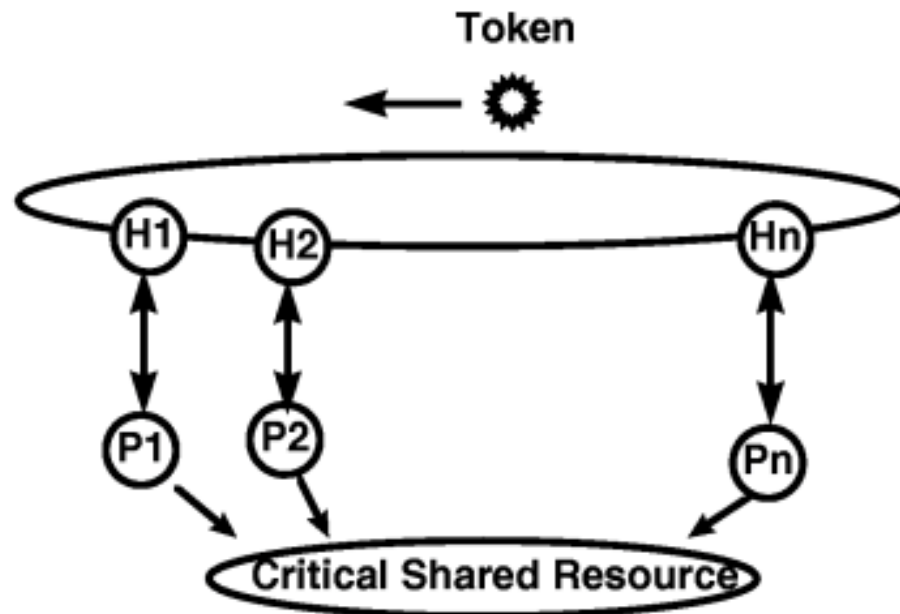
In telecommunication, token passing is a channel access method where a signal called a token is passed between nodes that authorizes the node to communicate. The most well-known examples are token ring and ARCNET.



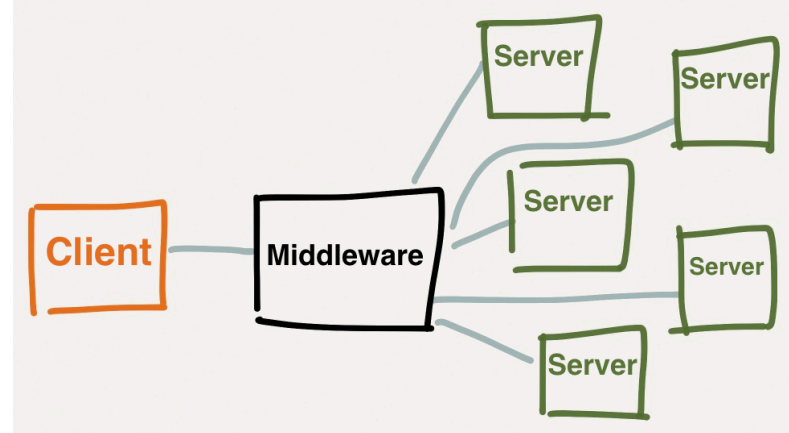
# Token passing - Distribuované zámky

## • Detekce ukončení procesů

- Token cirkuluje
- Ukončený proces produkuje token a předá
- Pokud process obdrží token, dokončí úlohu a předá token
- Pokud process co vyslal token tentýž obdrží ví ze je úloha ukončena

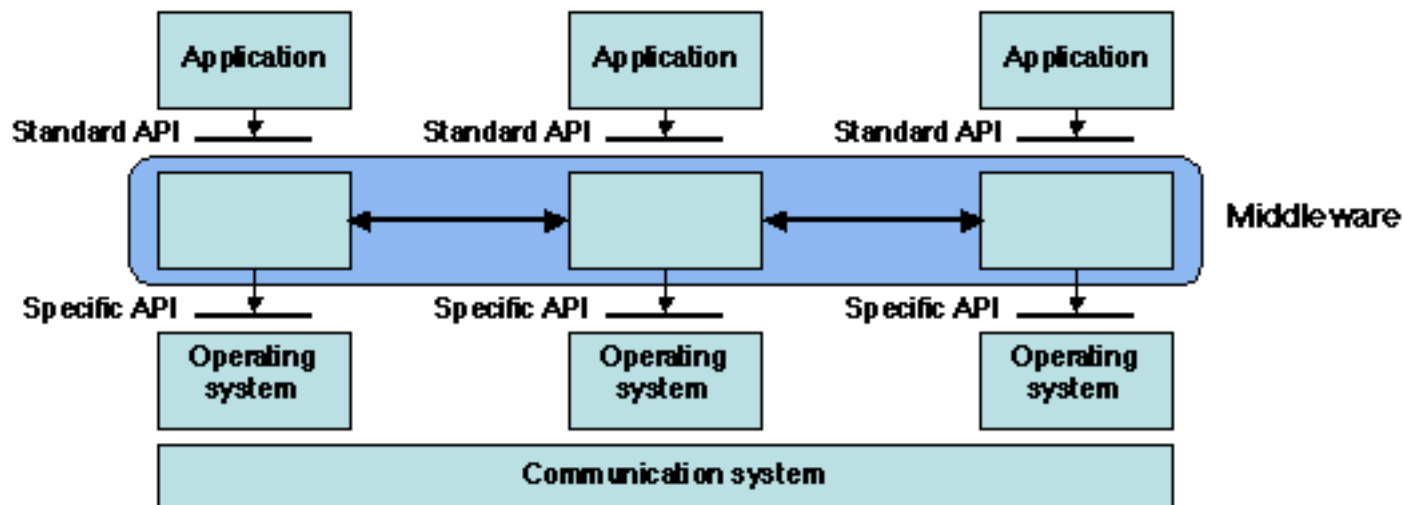


# Token passing



- **Middleware**

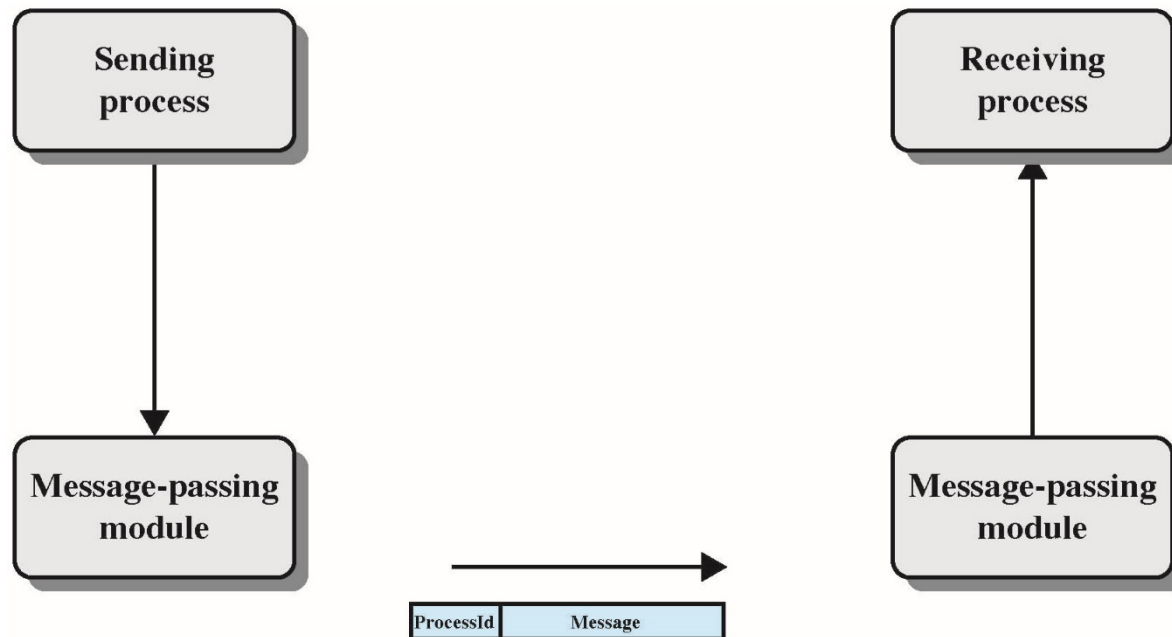
**Middleware** is the software that connects software components or enterprise applications. **Middleware** is the software layer that lies between the operating system and the applications on each side of a distributed computer network. Typically, it supports complex, distributed business software applications.



# Token passing

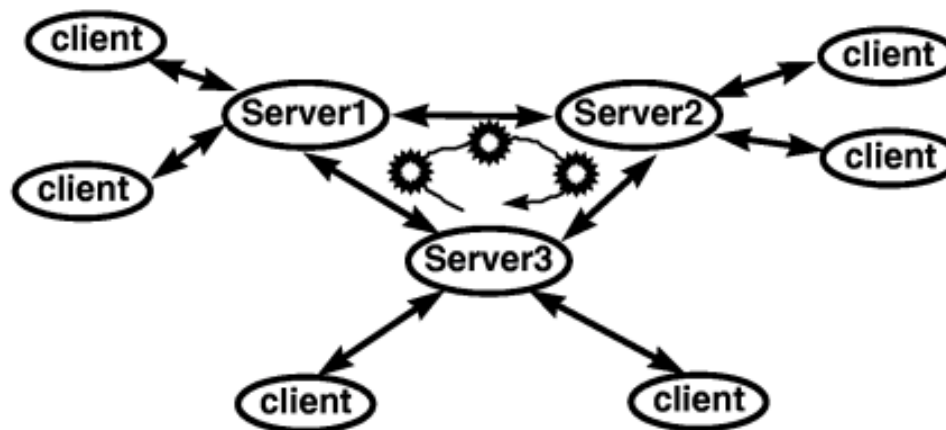
- Distributed Message Passing
- **Middleware** products are typically based on one of two underlying mechanisms:

*Message-passing* or **RPC** (Remote procedure calls)



# Replikace serverů

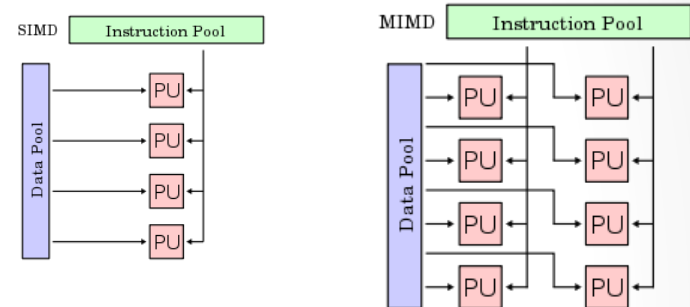
- Replikace serverů s identickou funkcí interagují
  - Zvýšení dostupnosti služeb
  - Konkurenční zpracování
  - Sdílení úloh
  - Paralelní rozdělení a panuj
- Lze replikovat identická data (problém u změn)
- Dostupnost dat



# Replikace pracovníků, sdílené úkoly

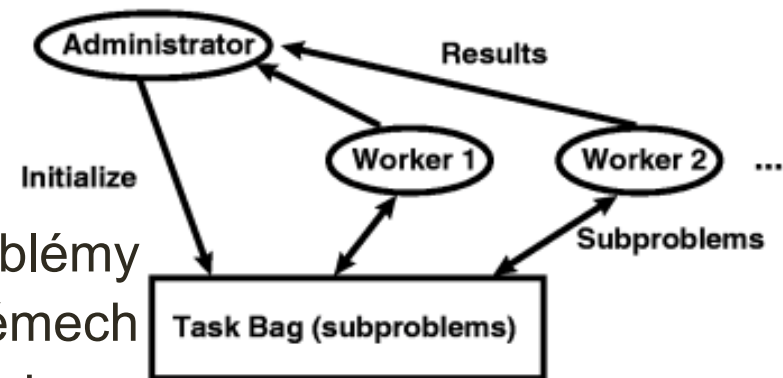
Tento styl má mnoho kopií výpočetních elementů

- Replikace pracantů
- Nezávislé podproblémy
  - Symmetric multiprocessing
  - Cluster
    - Propojené spolupracující uzly
    - Iluze jednoho stroje
    - Snadná škálovatelnost



## Parallel divide & conquer

- Jeden proces administrátor
  - řeší inicializaci a přiřazuje podproblémy
- Ostatní procesy pracující na podproblémech
- Pod-řešení probublají k administrátorovi
- Administrátor sestaví globální řešení



# Cloud vs Cluster

Platform as a service (PaaS) for cloud applications

- Scalable cluster of services

PaaS providers

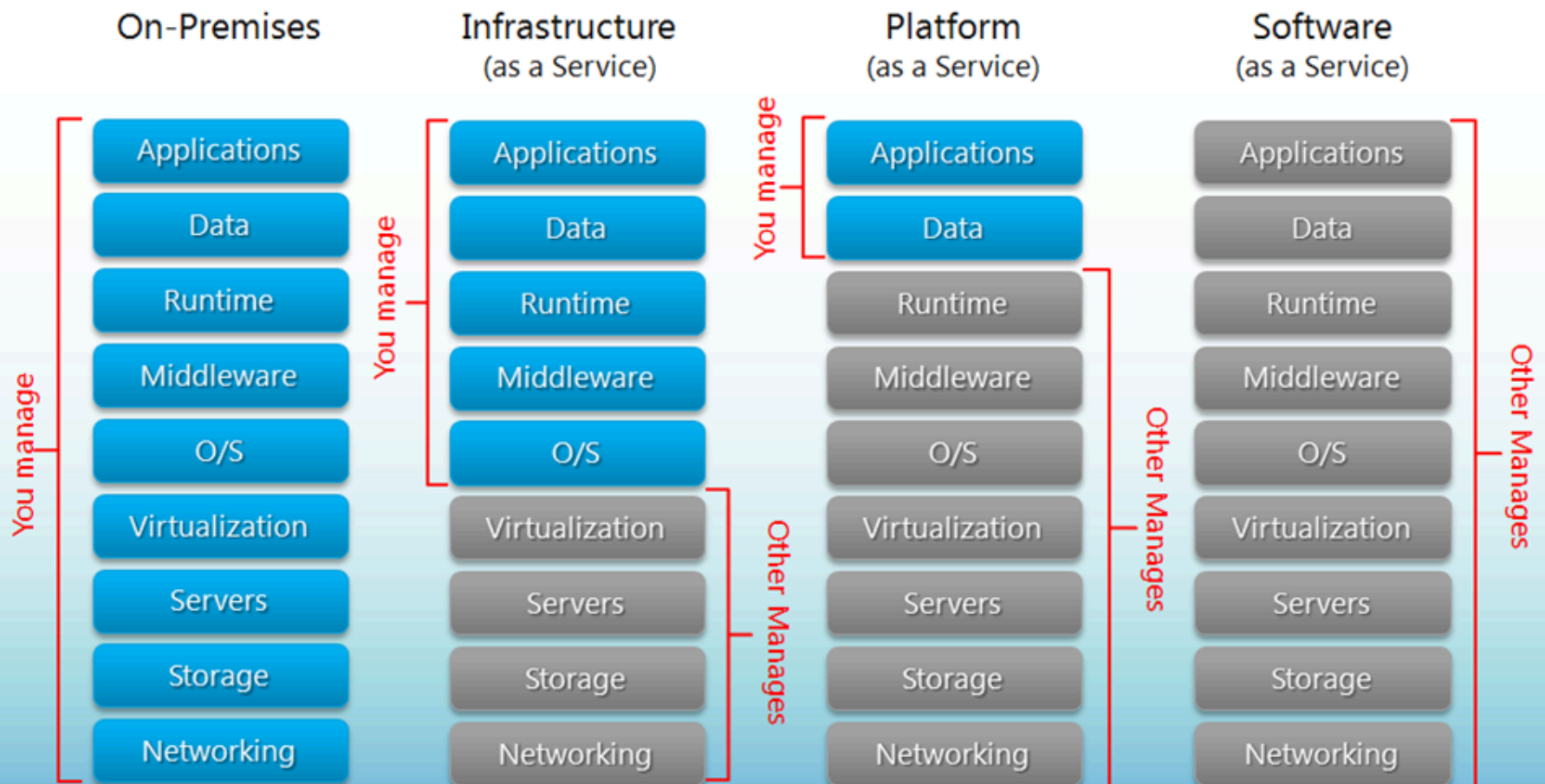
- Enable developers to deploy apps across clouds providing easily load balancing, scalability, high availability...
- Without the huge costs of buying, managing and maintaining their own environments.





# IaaS / PaaS / SaaS

## Separation of Responsibilities



# Decentralizace serverů vs replikace pracantů

- **Replikace**

- Hierarchická a synchronní struktura řízení
- Master node
  - Master-slave
  - Balancer

- **Decentralizace**

- Libovolná topologie řízení, asynchronní
- Žádný master

# Decentralizace serverů

- **Cloud**

- Service provided by single vendor
- Trust servers
- Client assumes single endpoint interaction

- **Peer to peer**

- Multiple vendors
  - Usually not certified
- No trust at all
- Dynamically changing environment

# Decentralizace serverů

## • Cloud computing

- The practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer.
  - PaaS

## • Peer to peer

- In its simplest form, a P2P network is created when two or more PCs are connected and share resources without going through a separate server computer.
- A P2P network can be an ad hoc connection - a couple of computers connected via a Universal Serial Bus to transfer files.

# Decentralizace serverů

- Peer-to-peer
- Uzel vlastnosti klient-server vlastní autonomní řízení
- Komu věřit a komu ne?



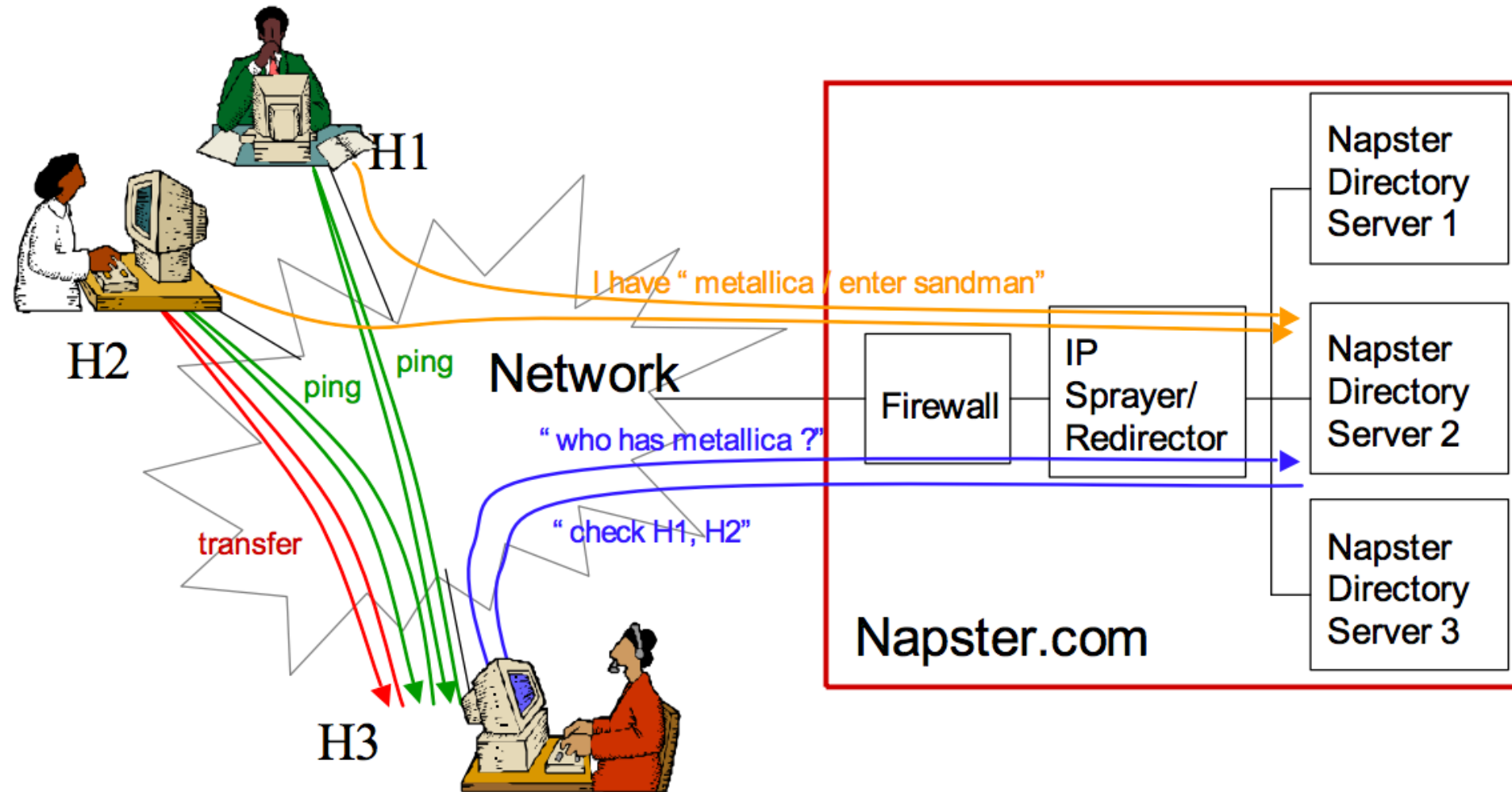
Server-based



P2P-network

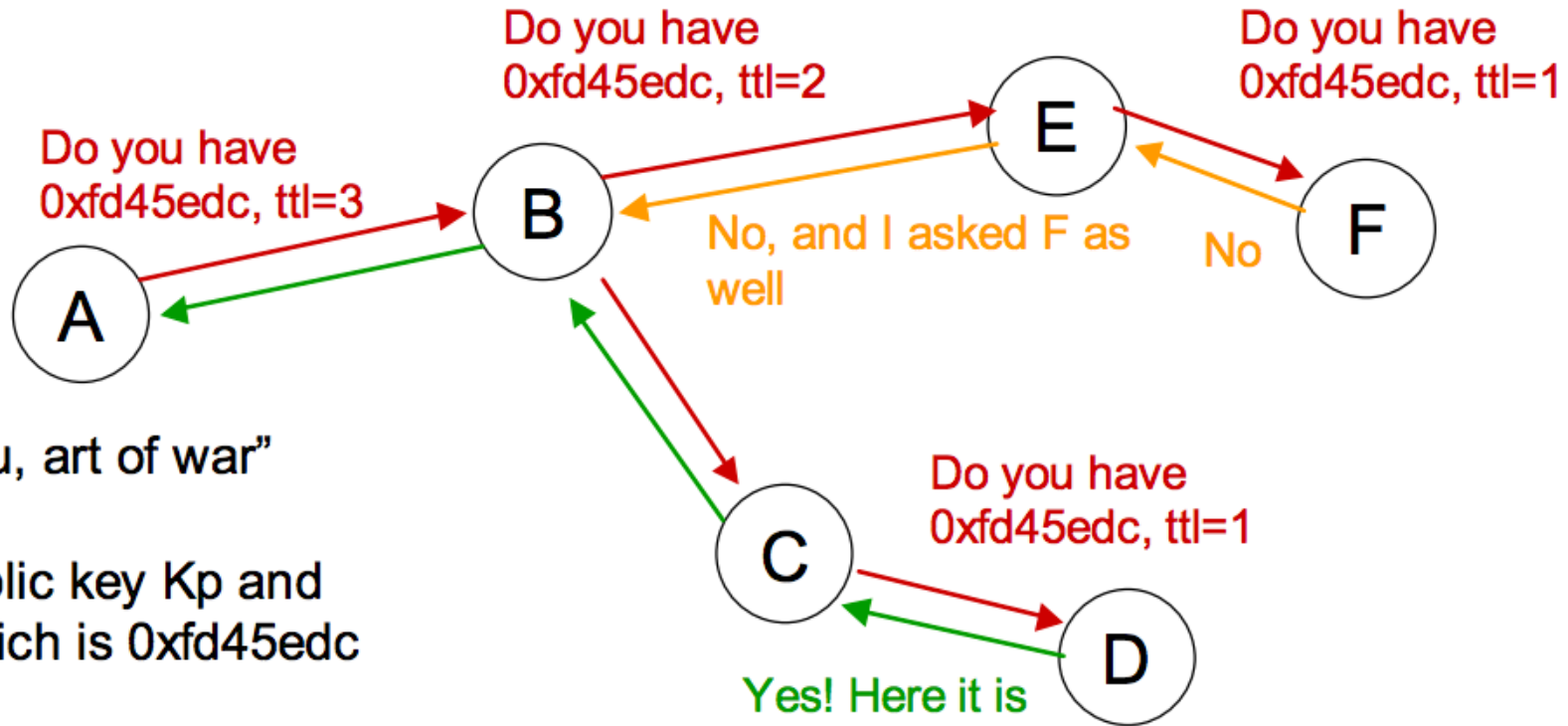
# Decentralize serverü

- Peer-to-peer – history connected with Napster



# Decentralize server

- Peer-to-peer – Solution with Freenet



Want "sun tzu, art of war"

Compute public key  $K_p$  and  
Hash( $K_p$ ) which is 0xfd45edc

Asks B for file with hash  
0xfd45edc, ttl=4

B asks E, then C

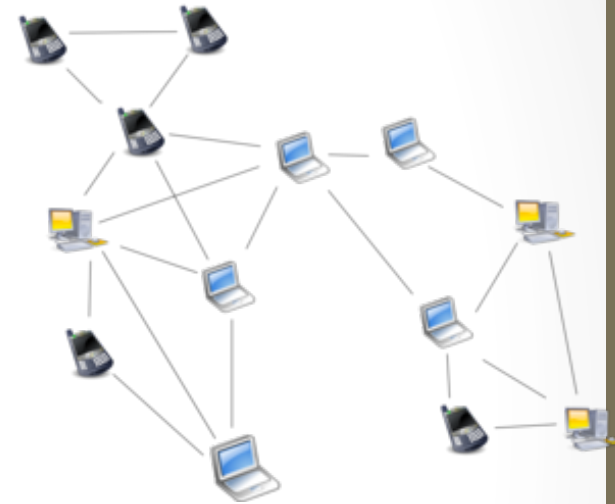
C retrieves it from D

A caches file

# Peer-to-peer Unstructured vs Structured networks

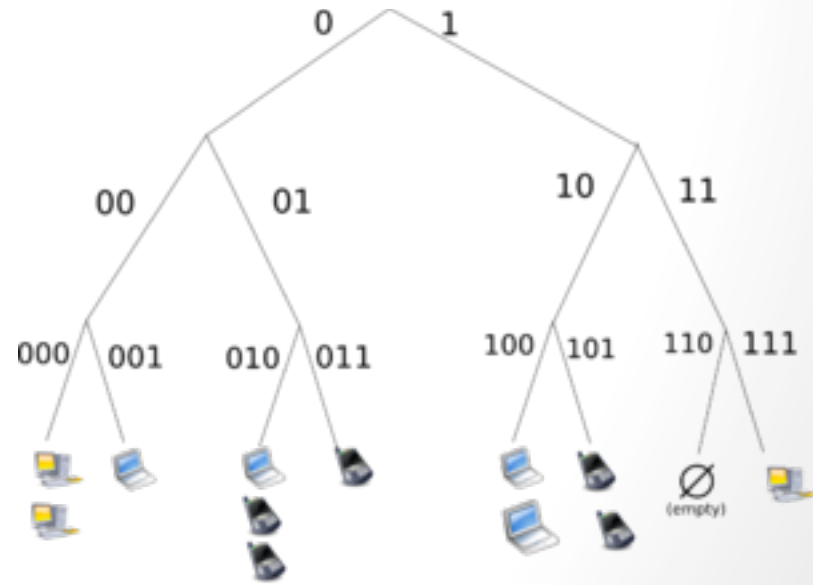
- **Unstructured**

- **Gossip**
- Gnutella
- Kazaa



- **Structured**

- **Distributed Hash Table**
- **Virtual overlays**
  - **Tree**
- Bittorrent
- Kademlia
- Chord

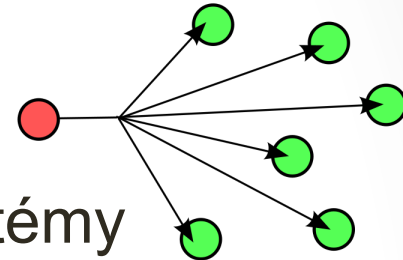




# \*cast

- Broadcast

- Pošly zprávy všem
- Př. Centrální hodiny pro real-time systémy



- Podskupiny

- Unicast, multicast, anycast, geocast..

