Chapter 1. Introduction

① **distributed system**: hardware or software components located at networked computers communicated and coordinate their actions only by passing message

② Significant consequences:

1) **concurrency**: 并发, concurrent program

2) **No global clock**: close coordination depends on a shared idea of the time at which the program actions occur

3) **Independent failures**: Each component of the system can fail independently, others still running.

③ **prime motivation**: to share resource

{ hardware: disk, printer
{ software: files, database ...

④ **Examples of DS**

Web search: Google

Massively multiplayer online games (MMOGs)
: the need for fast response time
to preserve the user experience

EVE: Client-Server

Financial trading: emphasis: { communication / processing

events

⑤ **Trends in DS**

{ pervasive networking: 泛在联网
ubiquitous computing: 无处不在计算
⇒ support user mobility.
~~#~~ demand for multimedia services: 多媒体
the view of ds as a utility: 设施

1) **Pervasive networking**

devices can be connected at any time and in any place

firewall: is to protect an ~~intranet~~ intranet by preventing unauthorized messages from leaving or entering, filtering incoming and outgoing message

ISP: Internet Service Providers, provide broadband links and other connections to individual users, enabling them to access services anywhere in the Internet

backbones: 主干网. with high transmission capacity, employing satellite connections, high-bandwith circuits.

2) **Mobile and ubiquitous computing**

~~mobile computing~~: users who are away from the home "Intranet" are still provided with access to resources via devices they carry.

ubiquitous computing: physical environments, home, office.
⇒ suggest small computing devices will eventually became pervasive in everyday objects.

Example: vertical washing machine

Three ways wireless connection:

{ laptop: wireless LAN.
mobile: 3G network, GPS.
digital camera: communicate over a personal wireless network. Printers.

⇒ spontaneous interoperation 自发互操作
service discovery.

3) Distributed multimedia system

a wide range of new services and apps can be provided on the desktop, including tv broadcasts, video, and io...

Weblasing

demands on underlying distributed infrastructure

1). encoding and encryption formats
2) to ensure the desired quality of service can be met
3) provide resource mangement strategy
4) provide adaptation strategy to deal with the situation where qualty service cannot be met.

4) Distributed computing as a utility 按需供给

users may actualy be provided with services by a virtual rather than a physical node. greater flexibility

cloud computing, cluster computers 集群计算机

to provide necessary scale and performance

Overal goal: to provide a range of cloud services, including high-performance computer capabilities, mass storage, other application services.

6 Resource Sharing

to reduce costs.

Service: file service. provide: read, write, delete operations.

Server: a running program on a networked computer that accepts requests from program and responds appropriately.

A complete interaction between client and server, called.

remote invocation

client server refers to processes, not computer

7 Challenges

1) Heterogeneity: 异构性

- networks: Internet protocols
- computer hardware: data types, integers, different ways: Unix, windows
- operating system
- programming languages: array, records
- implementations by different developers.

standards

middleware: 屏蔽 masking the heterogeneity of networks, hardware, operating system, programming language. CORBA

Java Remote Method Invocation RMI

middleware is implemented in Internet protocols.

middleware provids a uniform computational model including remote object invocation, remote event notification, remote SQL access, distributed transaction processing.

Send sensitive information.
↓ 措施
① Concealing the content.
② Knowing for sure the identity of the user.

Mobile code : virtual machine.
Java virtual machine (JVM).

Challenges
1) Denial of service attacks.
拒绝服务攻击.
2) Security of mobile code.

2) Openness 开放性.
⇒ Key interfaces are published.
RFCs, (Requests for Comments), for Internet protocols design.

4) Scalability. 伸缩性.
it will remain effective when there is a significant increase in the number of resources and users.
1 server → 20 users
2 server → 40 users.

Challenges
① controlling the cost of physical resources.
1→2 20→40
② Controlling the performance cost
hierarchic structures scale better than linear structure
time O(log n), ⇒ performance loss.
③ Preventing the software resources running out.
如IP address, 32 bits.
⇒ 128 bits.

Summarize:
① the fact: key interfaces are published.
② open ds are based on the uniform communication mechanism and published interfaces to share resources.
③ open ds can be constructed from heterogeneous hardware and software, but must be carefully tested.

3) Security three components.
① Confidentiality: 机密性 (protection against disclosure to unauthorized individuals)
② Integrity (完整性), 修改 alteration or corruption. 破坏
③ availability (可用性): interference with the means to access the resources.

④ avoiding performance bottlenecks: 性能瓶颈.

⑤ Failure handling: partial, some fails, with others continue

① detecting failures: some could be detected. [checksums]

② Masking failures: some failures can be hidden or made less severe. For example: (1) message can be retransmitted when they fail to arrive.

(2) File data can be written to a pair of disks.

③ Tolerating failures: web server↓. user can't wait forever, => inform the problem, try again later.

④ Recover from failures: [rolled back].

⑤ Redundancy: redundant components. database will be replicated in several servers.

⑥ Concurrency 并发.

Several users access a shared resource at the same time.

[throughput] 吞吐量.

⑦ [transparency] 透明性.

8. forms.

① [access transparency]: enables local and remote resources to be accessed using identical operations.

② [Location ...]: without physical location.

③ [concurrency ...]: several processes to operate concurrently using shared resources without interference

④ [Replication ...]: 使用监视多个副本. 提升性能.

⑤ [Failure ...] user can complete their task despite failures.

⑥ [Mobility ...], movements of resources and users without affecting the operation

⑦ [performance ...], allow the system to be [reconfigured] to improve performance as loads vary.

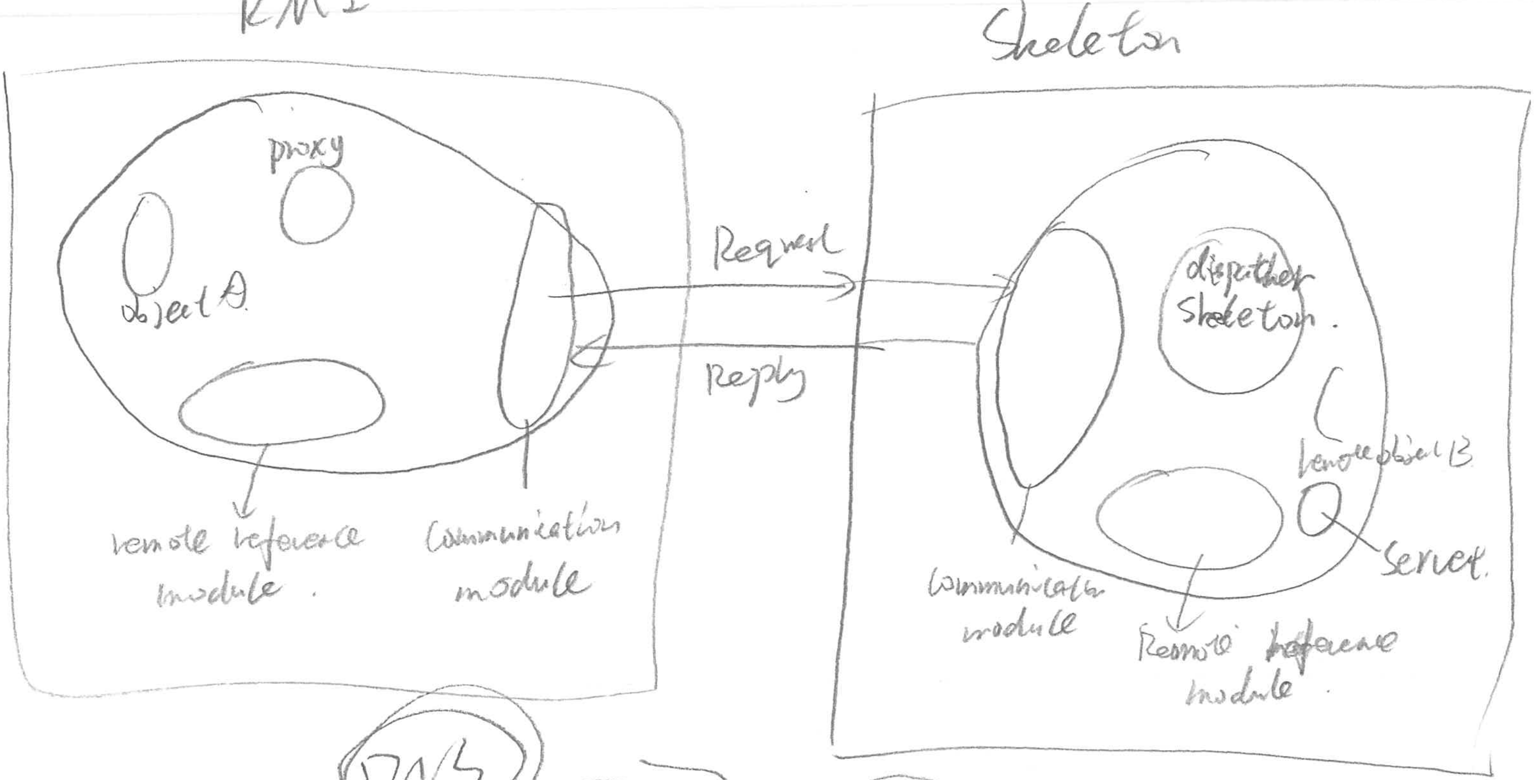
⑧ [Scaling ...]: can expand in system structure scale without changing

[access and locate transparency]

=> network transparency.

⑧ [Quality of services]

RMI



DNS → eventual consistency：每个节点是available

任何写操作，后台同步给其他节点 → 在任意时候，inconsistent 概率，是很高的。
(P)

CAP.

PACELC

一致性
consistent：当写操作完成后，之后的读写都得到最新值 eventual

可用性
available：只有一个节点服务器在工作，整个系统对于请求总是可以返回的

partition tolerant：当节点之间通信中断，整个系统仍然能提供服务。

It states that in case of network partitioning in a d s, one has to chose between availability (A) and consistency (C) (as per the CAP theorem)

~~H~~tolerates latency and faults

but else (E), even when the system is running normally in the absence of partitions, one has to chose between latency (L) and consistency (C)

Broker

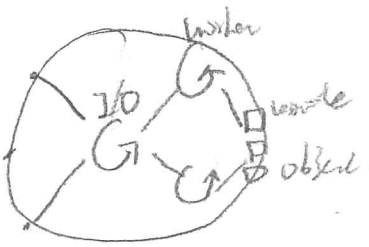
Kafka. ~~位~~ topic based.

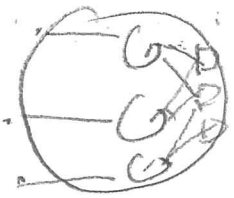
① 日志收集.
② 用户活动跟踪：用户的活动，topic，订阅topic，监控分析.
③ 运营指标.

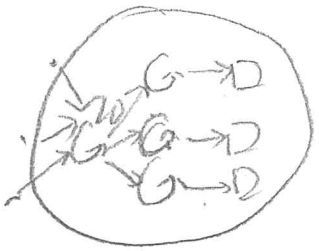
high throughput. low latency.
scalability：cluster
persistent：消息 → 持久化存储，...
fault tolerate：允许集群中节点失败.
高并发：支持多客户端同时读写

access control ?

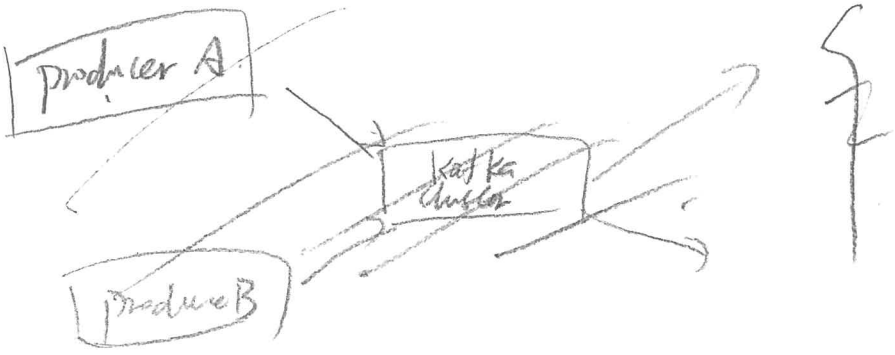
Thread - per - request · Thread per connect · Thread per objec

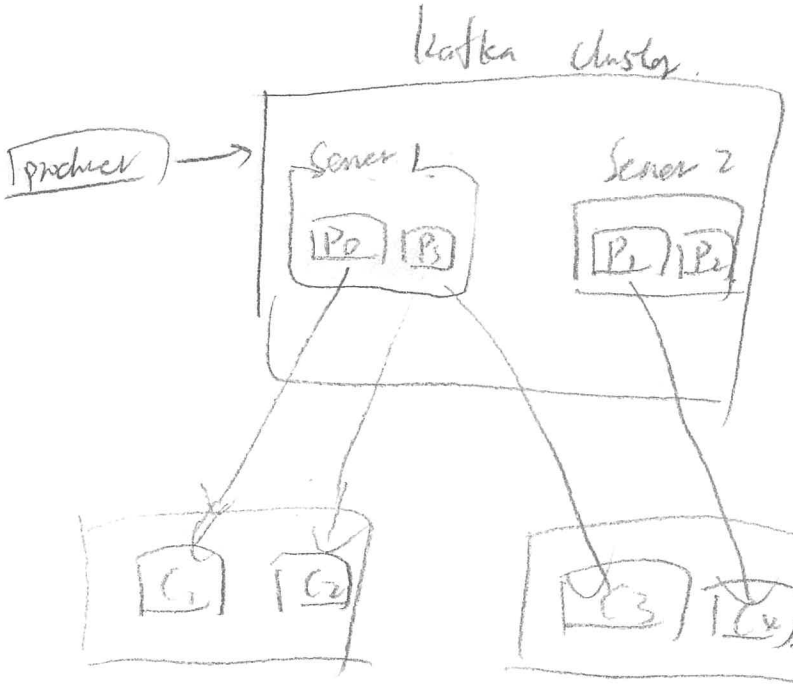


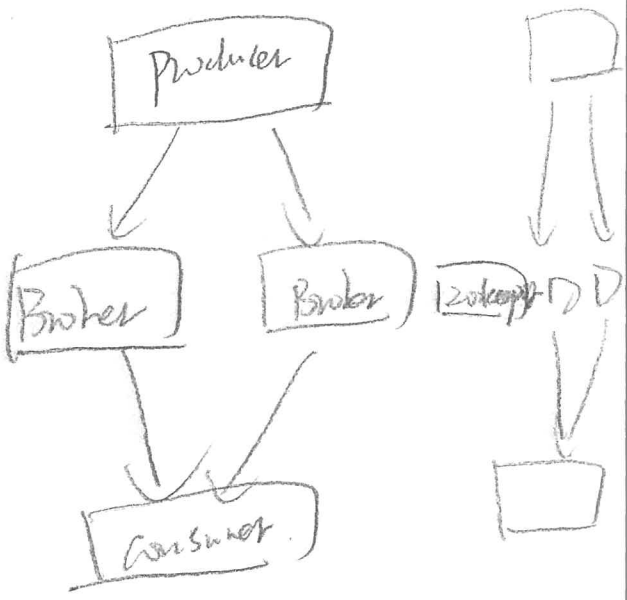




idempotent







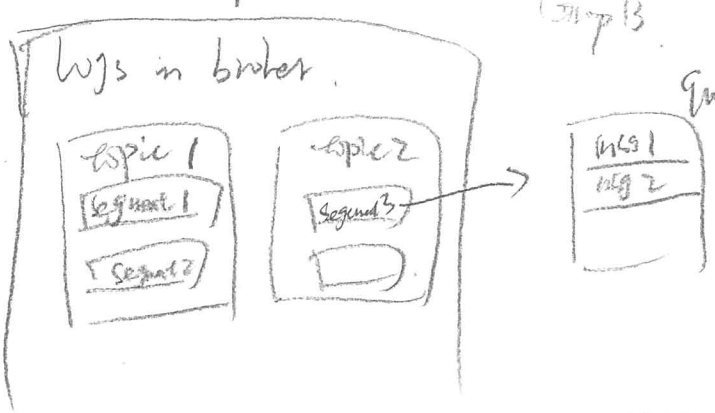
① topic has partitions.

② partition has a log on disk.

message persisted in log

Consumer Group A

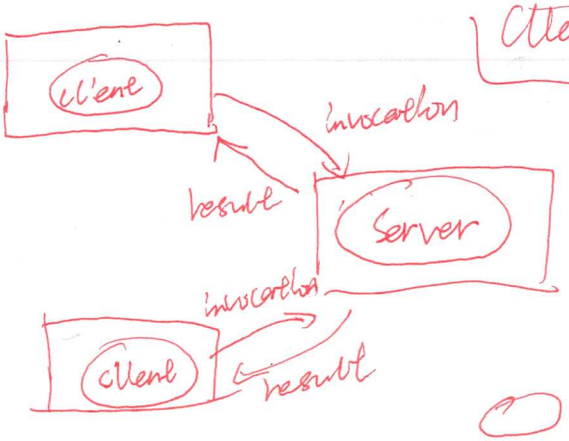
Group B



if broker down

⇒ { partitions unavailable
permanent data loss

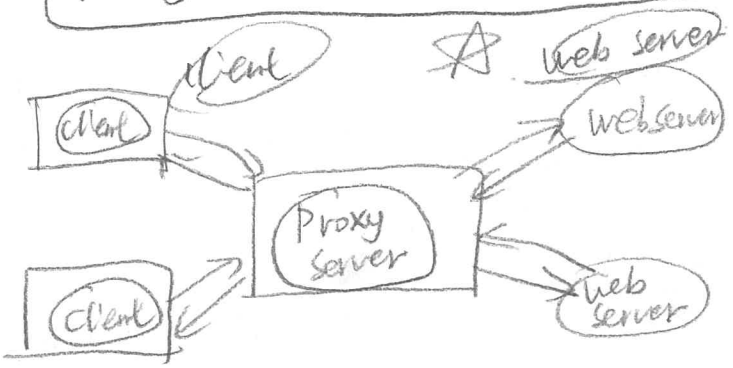
Client-Server



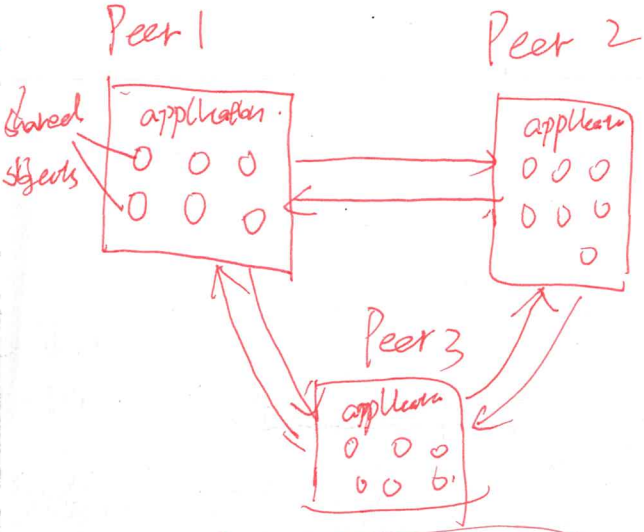
: process

: computer

Proxy servers and caches

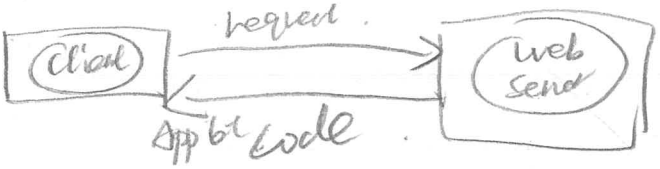


Peer-to-Peer



Mobile code

a) client request result from web



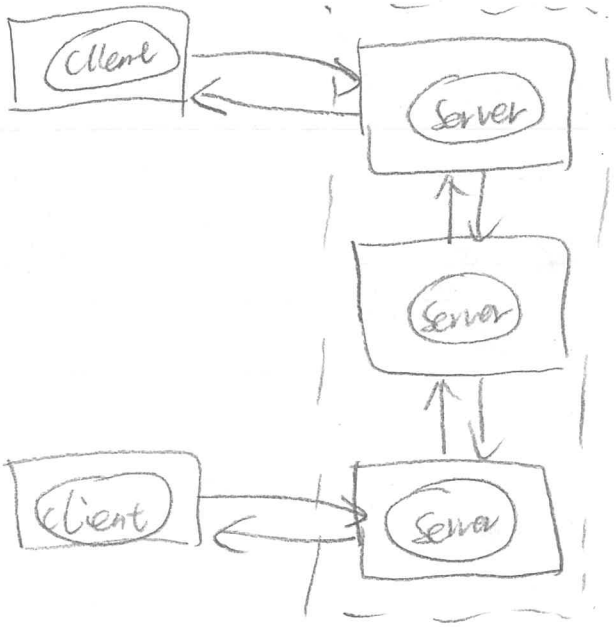
b) client interacts applet.

Thin Client

Software not downloaded; run on the server

Network computer or PC

Service



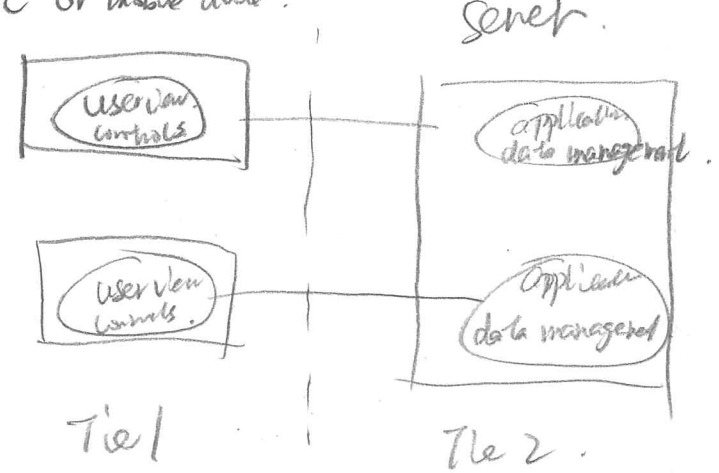
⇒ improve performance and reliability

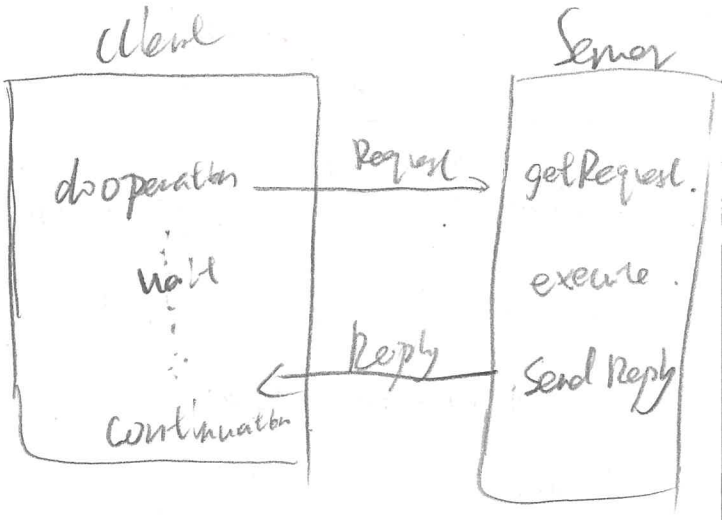
① Service is implemented as several server processes in separate host computers interacting as necessary to provide a service to client process.

Tired architecture

Request – Reply protocol

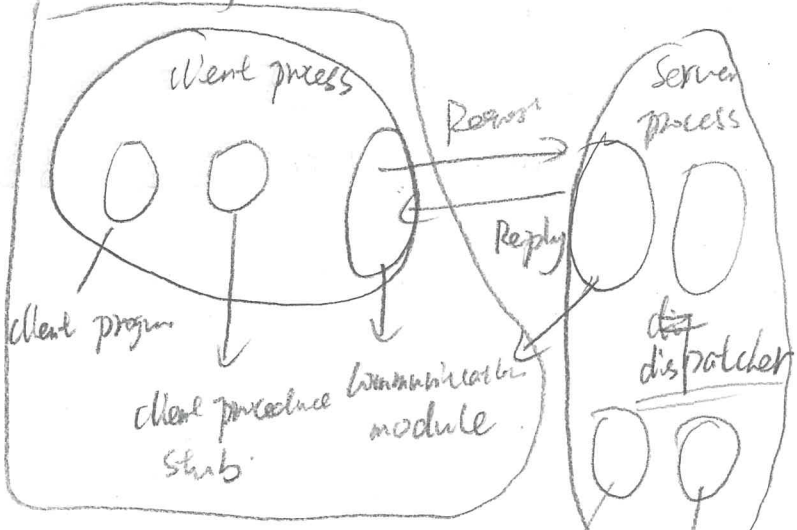
a) PC or mobile device.





b) PC or mobile devices

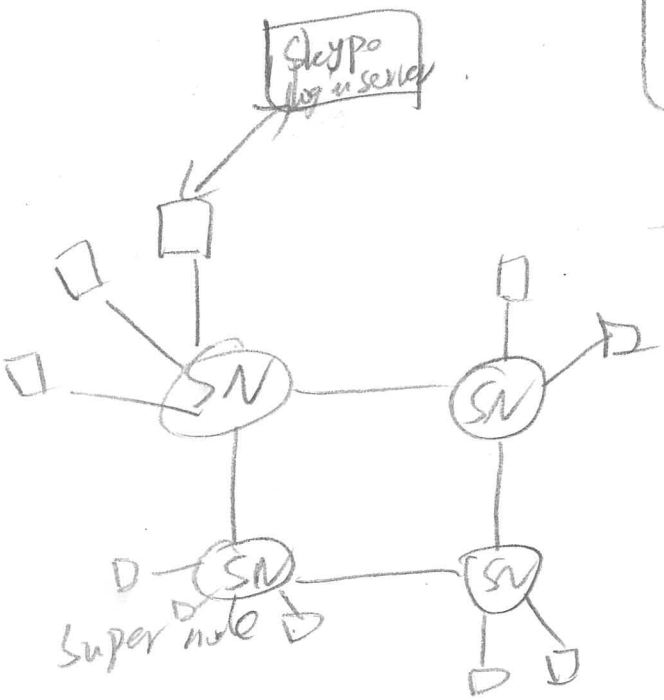
RPC



Overlay networks

SysKype

Skype login server



procedure identifiers

Service procedure