Distributed Systems

Summary

Prof. Dr. Oliver Hahm

Frankfurt University of Applied Sciences
Faculty 2: Computer Science and Engineering
oliver.hahm@fb2.fra-uas.de
https://teaching.dahahm.de

07.07.2024

Agenda

- Exam
- Key Terms
- Questions

Agenda

- Exam
- Key Terms
- Questions

Current Situation

- Until further notice you can contact me via
  `mailto:teaching@dahahm.de`
  or
  `mailto:oliver.hahm@informatik-ffm.de`
- Gitlab is accessible from within the university network (VPN is **not** possible)
- Alternatively you can submit your solutions via mail
- For urgent matters there are some additional office hours (see `https://teaching.dahahm.de/news`

Organizational

- The exam will take place on July 31, 2024
- You will be allowed to bring a (self-written or printed) cheat sheet and a calculator
- *Be there on time!*

Content

- All necessary formulas, concrete numbers, and potential APIs will be given in the exam
- The exam will consist of similar tasks and look similar to the mock exam

Reminder

- What is necessary to pass the exam?
- You should be able to ...
  - explain main concepts and ideas with your own words,
  - select a suitable solution for a given problem,
  - analyze a given solution and detect (potential) problems, and
  - explain your answers.

Agenda

- Exam
- Key Terms
- Questions

Key Terms (1/5)

- Distributed system, distributed application/program

Key Terms (1/5)

- Distributed system, distributed application/program
- Strong and loose coupling

Key Terms (1/5)

- Distributed system, distributed application/program
- Strong and loose coupling
- Types of transparency:

Key Terms (1/5)

- Distributed system, distributed application/program
- Strong and loose coupling
- Types of transparency:
  - Location transparency

Key Terms (1/5)

- Distributed system, distributed application/program
- Strong and loose coupling
- Types of transparency:
  - Location transparency
  - Access transparency

Key Terms (1/5)

- Distributed system, distributed application/program
- Strong and loose coupling
- Types of transparency:
  - Location transparency
  - Access transparency
  - Migration/mobility transparency

Key Terms (1/5)

- Distributed system, distributed application/program
- Strong and loose coupling
- Types of transparency:
  - Location transparency
  - Access transparency
  - Migration/mobility transparency
  - Replication transparency

Key Terms (1/5)

- Distributed system, distributed application/program
- Strong and loose coupling
- Types of transparency:
  - Location transparency
  - Access transparency
  - Migration/mobility transparency
  - Replication transparency
  - Concurrency transparency

Key Terms (1/5)

- Distributed system, distributed application/program
- Strong and loose coupling
- Types of transparency:
  - Location transparency
  - Access transparency
  - Migration/mobility transparency
  - Replication transparency
  - Concurrency transparency
  - Scaling transparency

Key Terms (1/5)

- Distributed system, distributed application/program
- Strong and loose coupling
- Types of transparency:
  - Location transparency
  - Access transparency
  - Migration/mobility transparency
  - Replication transparency
  - Concurrency transparency
  - Scaling transparency
  - Performance transparency

Key Terms (1/5)

- Distributed system, distributed application/program
- Strong and loose coupling
- Types of transparency:
  - Location transparency
  - Access transparency
  - Migration/mobility transparency
  - Replication transparency
  - Concurrency transparency
  - Scaling transparency
  - Performance transparency
  - Failure transparency

Key Terms (1/5)

- Distributed system, distributed application/program
- Strong and loose coupling
- Types of transparency:
  - Location transparency
  - Access transparency
  - Migration/mobility transparency
  - Replication transparency
  - Concurrency transparency
  - Scaling transparency
  - Performance transparency
  - Failure transparency
- Robustness

Key Terms (1/5)

- Distributed system, distributed application/program
- Strong and loose coupling
- Types of transparency:
  - Location transparency
  - Access transparency
  - Migration/mobility transparency
  - Replication transparency
  - Concurrency transparency
  - Scaling transparency
  - Performance transparency
  - Failure transparency
- Robustness
- Scalability

Key Terms (1/5)

- Distributed system, distributed application/program
- Strong and loose coupling
- Types of transparency:
  - Location transparency
  - Access transparency
  - Migration/mobility transparency
  - Replication transparency
  - Concurrency transparency
  - Scaling transparency
  - Performance transparency
  - Failure transparency
- Robustness
- Scalability
- Openness

Key Terms (2/5)

- Layers of the OSI reference model

Key Terms (2/5)

- Layers of the OSI reference model
- Directed vs. undirected communication channels

Key Terms (2/5)

- Layers of the OSI reference model
- Directed vs. undirected communication channels
- Direct/indirect and symmetrical/asymmetrical addressing

Key Terms (2/5)

- Layers of the OSI reference model
- Directed vs. undirected communication channels
- Direct/indirect and symmetrical/asymmetrical addressing
- Channel capacity

Key Terms (2/5)

- Layers of the OSI reference model
- Directed vs. undirected communication channels
- Direct/indirect and symmetrical/asymmetrical addressing
- Channel capacity
- Communication patterns
  - One-way
  - Request-response
  - Synchronous vs. asynchronous calls
  - Publish-subscribe

Key Terms (2/5)

- Layers of the OSI reference model
- Directed vs. undirected communication channels
- Direct/indirect and symmetrical/asymmetrical addressing
- Channel capacity
- Communication patterns
  - One-way
  - Request-response
  - Synchronous vs. asynchronous calls
  - Publish-subscribe
- Message semantics
  - Byte stream
  - Message container
  - Typed messages

Key Terms (3/5)

- Server architectures

Key Terms (3/5)

- Server architectures
- Sockets

Key Terms (3/5)

- Server architectures
- Sockets
- Inter-Process Communication

Key Terms (3/5)

- Server architectures
- Sockets
- Inter-Process Communication
- Network representations (XDR, CDR, ASN.1, JSON . . .)

Key Terms (3/5)

- Server architectures
- Sockets
- Inter-Process Communication
- Network representations (XDR, CDR, ASN.1, JSON . . .)
- Remote Procedure Calls (RPCs)
  - Binding
  - Stub
  - Interface Description Language (IDL)
  - Error semantics

Key Terms (3/5)

- Server architectures
- Sockets
- Inter-Process Communication
- Network representations (XDR, CDR, ASN.1, JSON . . .)
- Remote Procedure Calls (RPCs)
  - Binding
  - Stub
  - Interface Description Language (IDL)
  - Error semantics
- Names, their properties, and name spaces

Key Terms (3/5)

- Server architectures
- Sockets
- Inter-Process Communication
- Network representations (XDR, CDR, ASN.1, JSON . . .)
- Remote Procedure Calls (RPCs)
  - Binding
  - Stub
  - Interface Description Language (IDL)
  - Error semantics
- Names, their properties, and name spaces
- Name and directory services

Key Terms (4/5)

- Distributed file systems
  - Consistency semantics
  - Stateful vs. stateless servers
  - NFS and AFS

Key Terms (4/5)

- Distributed file systems
  - Consistency semantics
  - Stateful vs. stateless servers
  - NFS and AFS
- Happened-before relationship

Key Terms (4/5)

- Distributed file systems
  - Consistency semantics
  - Stateful vs. stateless servers
  - NFS and AFS
- Happened-before relationship
- Global time and clock synchronization

Key Terms (4/5)

- Distributed file systems
  - Consistency semantics
  - Stateful vs. stateless servers
  - NFS and AFS
- Happened-before relationship
- Global time and clock synchronization
- Logical clocks
  - Lamport clocks
  - Vector clocks
  - Clock condition
  - Causality problem

Key Terms (5/5)

- RESTful APIs and web services

Key Terms (5/5)

- RESTful APIs and web services
- Security

Key Terms (5/5)

- RESTful APIs and web services
- Security
  - Protection goals

Key Terms (5/5)

- RESTful APIs and web services
- Security
  - Protection goals
  - Threats

Key Terms (5/5)

- RESTful APIs and web services
- Security
  - Protection goals
  - Threats
  - Symmetric and asymmetric cryptography

Key Terms (5/5)

- RESTful APIs and web services
- Security
  - Protection goals
  - Threats
  - Symmetric and asymmetric cryptography
  - Cryptographic hashing

Key Terms (5/5)

- RESTful APIs and web services
- Security
  - Protection goals
  - Threats
  - Symmetric and asymmetric cryptography
  - Cryptographic hashing
  - Authentication

Key Terms (5/5)

- RESTful APIs and web services
- Security
  - Protection goals
  - Threats
  - Symmetric and asymmetric cryptography
  - Cryptographic hashing
  - Authentication
  - Digital signatures

Key Terms (5/5)

- RESTful APIs and web services
- Security
  - Protection goals
  - Threats
  - Symmetric and asymmetric cryptography
  - Cryptographic hashing
  - Authentication
  - Digital signatures
  - PKI and TLS

Key Terms (5/5)

- RESTful APIs and web services
- Security
  - Protection goals
  - Threats
  - Symmetric and asymmetric cryptography
  - Cryptographic hashing
  - Authentication
  - Digital signatures
  - PKI and TLS
  - Firewalls

Agenda

- Exam
- Key Terms
- Questions

Network Operating Systeme

Describe the term *Network Operating Systems*

Network Operating Systeme

Describe the term *Network Operating Systems*

Answer

A network operating system is an operating system that provides certain functions in multicomputer operation more or less transparently, so that distributed applications can be implemented on it.

Server Architectures

Describe different alternatives for server architectures.

Server Architectures

Describe different alternatives for server architectures.

Answer

- Simple sequential process
- Parallel processes
- Sequential Server with state machine (`select`)
- Multithreaded Server

Rendez-Vous

Name the pros and cons for rendezvous communication.

Rendez-Vous

Name the pros and cons for rendezvous communication.

Answer

- Advantages
  - Sender knows that the message has been received
  - No additional memory for buffering required
- Drawbacks
  - inflexible
  - Synchronisation on terminal may be difficult
  - No buffering
  - Waiting for the slowest peer

Pub-Sub

Describe the Publisher/Subscriber model

Pub-Sub

Describe the Publisher/Subscriber model

Answer

- Messages categorized into topics (or event channels)
- Receiver (subscribe) subscribe to topics
- Sender (publisher) publish messages (or events)
- Model allows for transparent sending of a single message to multiple receivers
- Requires typically a broker as an intermediary instance

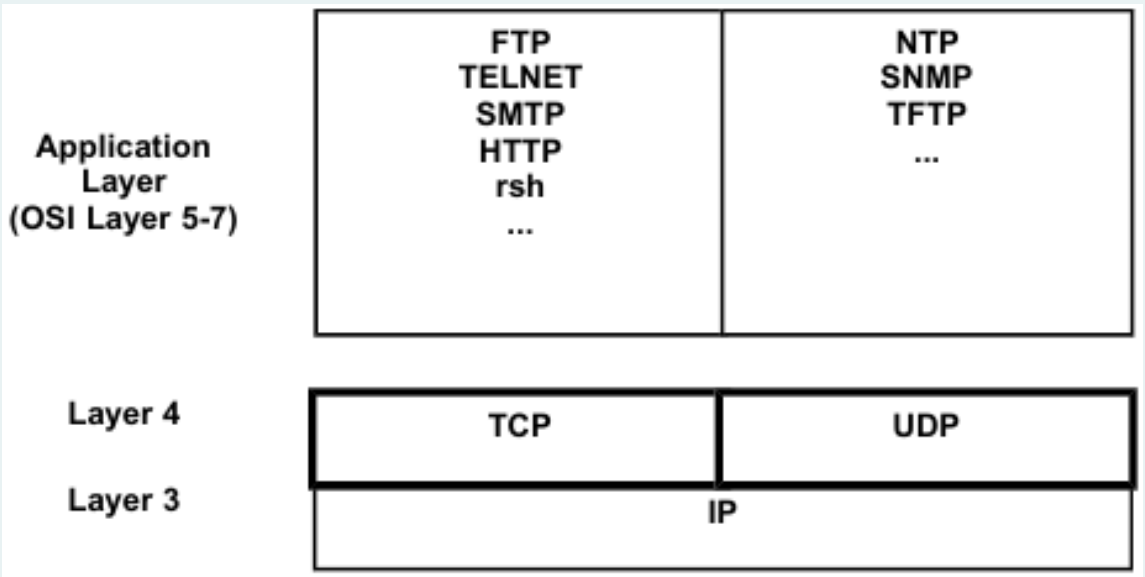
ISO/OSI Model

Name some examples for protocols from the TCP/IP family according to the OSI model.

ISO/OSI Model

Name some examples for protocols from the TCP/IP family according to the OSI model.

Answer



Client/Server Applications

Consider a socked based network program for the implementation of a distributed client/server application.

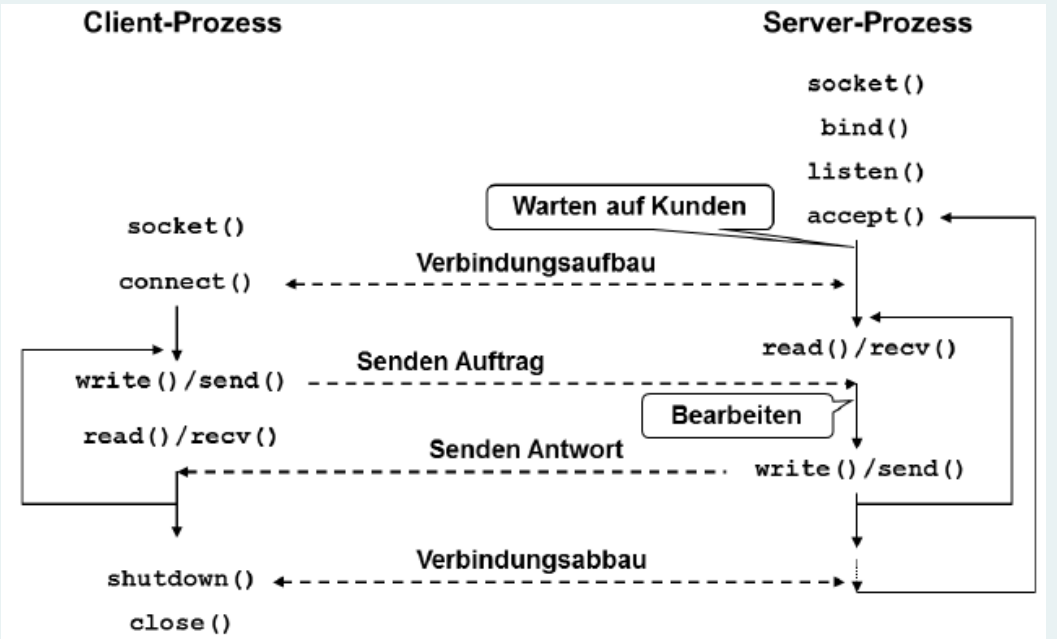
What is the general sequence and interaction of system calls on the client and server side if TCP is used as a transport layer protocol?

Client/Server Applications

Consider a socked based network program for the implementation of a distributed client/server application.

What is the general sequence and interaction of system calls on the client and server side if TCP is used as a transport layer protocol?

Answer



Network Data Presentation

A SPARC processor represents integer numbers as 32 bit words using the big endian format, an Intel x86 processor uses the little endian format. Which value do you receive on the x86 side, if a SPARC processor sends the value 3?

Which possibilities do exist to solve this issue?

Which layer of the OSI model handles this issue?

Network Data Presentation

A SPARC processor represents integer numbers as 32 bit words using the big endian format, an Intel x86 processor uses the little endian format. Which value do you receive on the x86 side, if a SPARC processor sends the value 3?

Answer

Swapping bytes transforms $0x00000003$ into
$0x03000000 = 2^{24} + 2^{25} = 50.331.648$

Which possibilities do exist to solve this issue?

Which layer of the OSI model handles this issue?

Network Data Presentation

A SPARC processor represents integer numbers as 32 bit words using the big endian format, an Intel x86 processor uses the little endian format. Which value do you receive on the x86 side, if a SPARC processor sends the value 3?

Answer

Swapping bytes transforms $0x00000003$ into
$0x03000000 = 2^{24} + 2^{25} = 50.331.648$

Which possibilities do exist to solve this issue?

Answer

Use network byte order, a mapping function, or a string format (e. g., JSON)

Which layer of the OSI model handles this issue?

Network Data Presentation

A SPARC processor represents integer numbers as 32 bit words using the big endian format, an Intel x86 processor uses the little endian format. Which value do you receive on the x86 side, if a SPARC processor sends the value 3?

Answer

Swapping bytes transforms $0x00000003$ into
$0x03000000 = 2^{24} + 2^{25} = 50.331.648$

Which possibilities do exist to solve this issue?

Answer

Use network byte order, a mapping function, or a string format (e. g., JSON)

Which layer of the OSI model handles this issue?

Answer

Presentation layer (OSI Layer 6)

Architecture Models

Describe different variants of the client/server model and explain the difference to the peer-to-peer model.

Architecture Models

Describe different variants of the client/server model and explain the difference to the peer-to-peer model.

Answer

- A service user (client) request at the service provider (server)
- A host can take the client and server role in parallel
- Communication via proxy is possible (proxy acts in both roles)
- For P2P: Communication between peers, no infrastructure required

Remote Procedure Calls

What is the definition of a Remote Procedure Call?

What does *binding* mean and which are the two sub-issues which have to be considered?

Remote Procedure Calls

What is the definition of a Remote Procedure Call?

Answer

RPC as a synchronous mechanism '"which transfers control flow and data as a procedure call between two address spaces over a narrowband network."'

What does *binding* mean and which are the two sub-issues which have to be considered?

Remote Procedure Calls

What is the definition of a Remote Procedure Call?

Answer

RPC as a synchronous mechanism '"which transfers control flow and data as a procedure call between two address spaces over a narrowband network."'

What does *binding* mean and which are the two sub-issues which have to be considered?

Answer

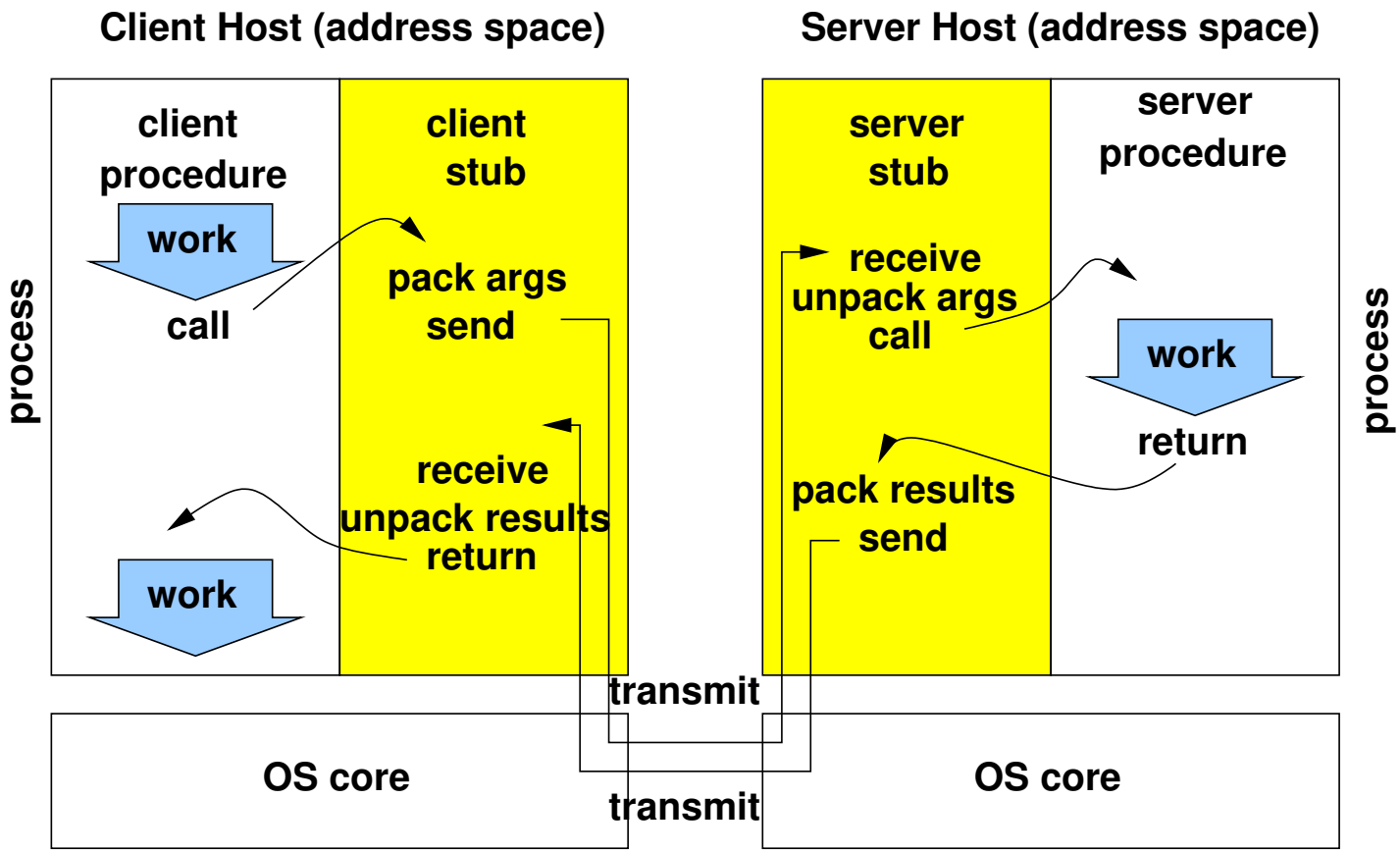
Binding binds a client to the server; Naming resolve the name and locating the address of the requested service.

Remote Procedure Calls (2)

Describe the main principle of an RPC. Which parts need to be implemented by the application developer?

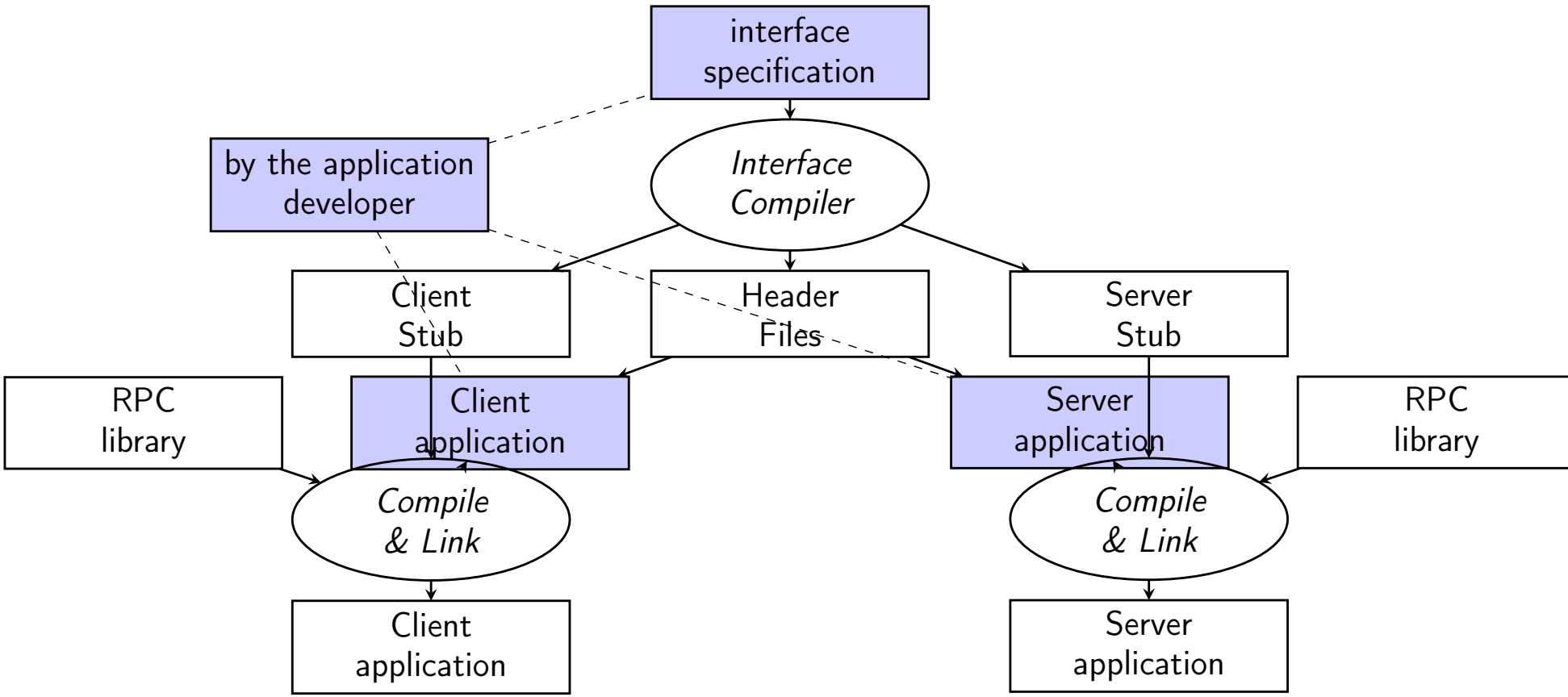
Remote Procedure Calls (2)

Describe the main principle of an RPC. Which parts need to be implemented by the application developer?



Remote Procedure Calls (2)

Describe the main principle of an RPC. Which parts need to be implemented by the application developer?



Remote Procedure Calls (3)

Name some popular examples for RPC systems.

Explain the difference between explicit and implicit typing and give an example for an according network representation format.

Remote Procedure Calls (3)

Name some popular examples for RPC systems.

Answer

Sun RPC, DCE-RPC, MS-RPC, Apache Thrift, XML-RPC, gRPC

Explain the difference between explicit and implicit typing and give an example for an according network representation format.

Remote Procedure Calls (3)

Name some popular examples for RPC systems.

Answer

Sun RPC, DCE-RPC, MS-RPC, Apache Thrift, XML-RPC, gRPC

Explain the difference between explicit and implicit typing and give an example for an according network representation format.

Answer

1. When using explicitly typed data:
   - data fields are preceded by their type information (self-describing)
   - Typical representation: TLV (Type/Length/Value)
   - Disadvantage: run time overhead
   - Example: ASN.1 BER
2. When using implicitly typed data:
   - Receiver has to know the type (e. g., from the interface description)
   - Example: SunRPC

Remote Procedure Calls (4)

Explain the problem when passing addresses in RPC calls and some solutions.

Remote Procedure Calls (4)

Explain the problem when passing addresses in RPC calls and some solutions.

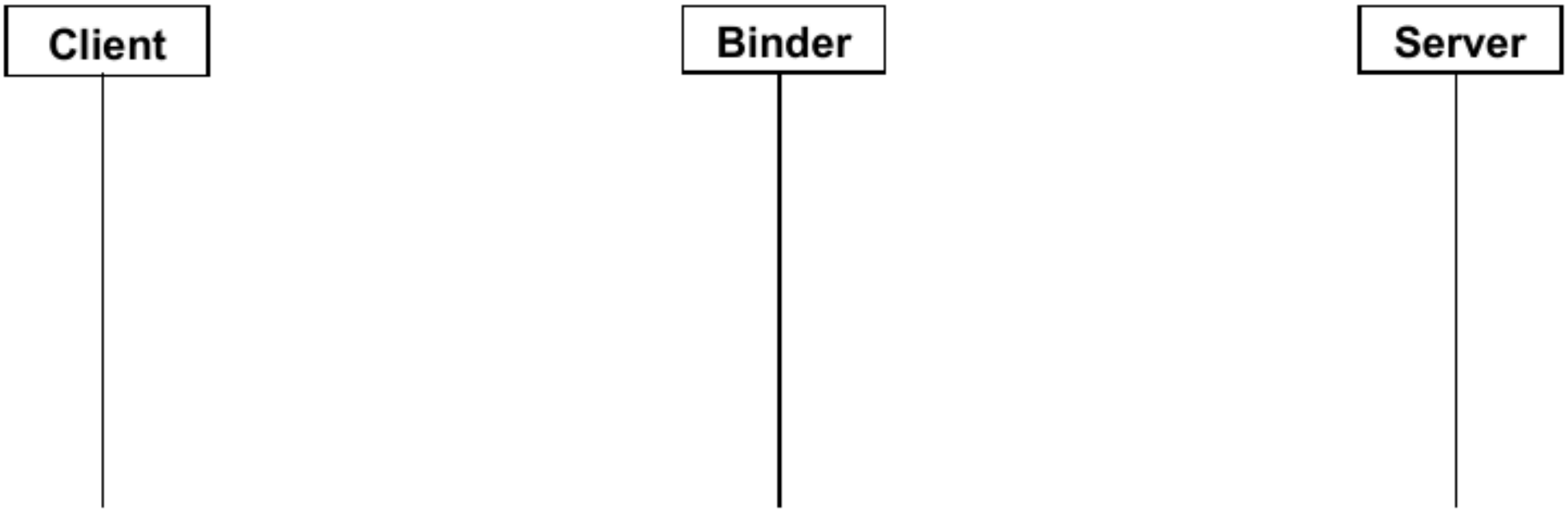
Answer

Address has no meaning in the destination address space.
Solutions:

- Forbid the use of pointers and references, only allow for call-by-value
- Use a shared, global address space if possible
- Replace pointers by markers and reconstruct compound data structures at receiver side with local pointers

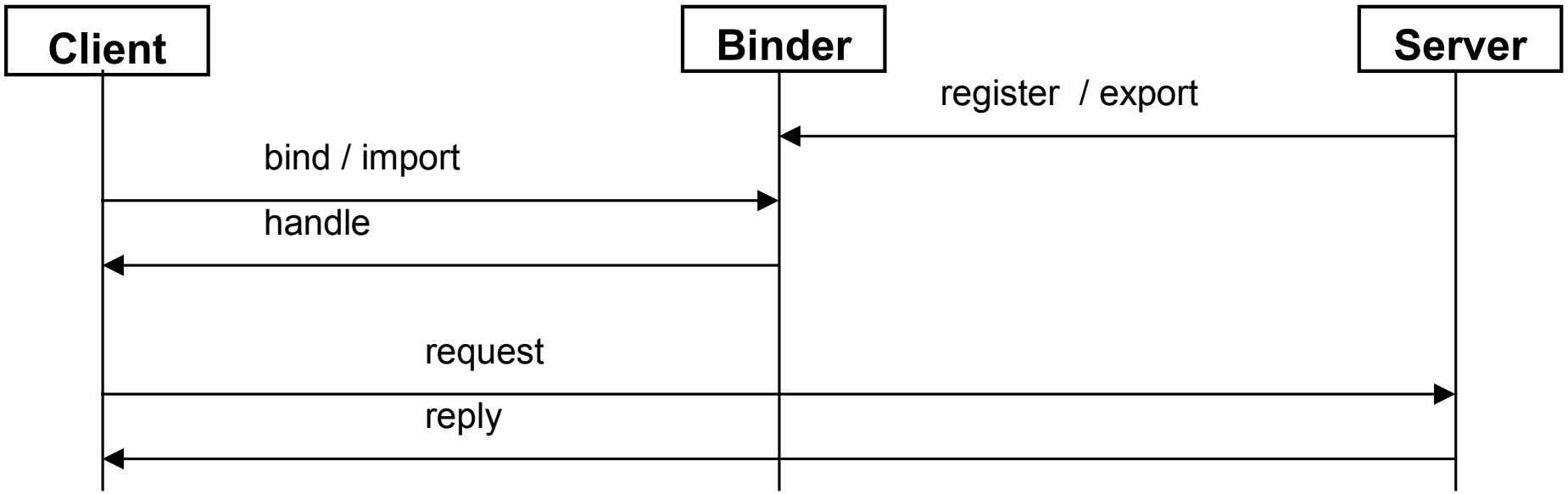
Binding in RPC Systems

Describe the binding sequence between client and server in an RPC system. Complete the following diagram with communication relationships and name their function.



Binding in RPC Systems

Describe the binding sequence between client and server in an RPC system. Complete the following diagram with communication relationships and name their function.



Error Semantics of RPCs

Describe the different semantics of RPCs in the case of an error.

Error Semantics of RPCs

Describe the different semantics of RPCs in the case of an error.

Answer

- *at-least-once* semantic
  - On successful RPC execution ⇒ called procedure gets executed at least once, i. e., multiple executions are possible
  - Any effect on error case is possible
  - In general only suited for idempotent operations, i. e., multiple calls does not change the state and result
  - If no result is received on the client side within a configured timeout, the call is repeated
  - No precautions on the server side are required
- *at-most-once* semantic
  - On successful RPC execution ⇒ called procedure gets executed exactly once
  - Non-successful RPC execution ⇒ called procedure gets never executed
  - No partial error consequences
  - More complex
  - Requires duplicate detection at the server
- *exactly-once* semantic
  - Like a local procedure call
  - Very complex

Name Properties

Describe the difference between pure and impure names. Give some examples

Name Properties

Describe the difference between pure and impure names. Give some examples

Answer

- Pure:
  - A name is only a bit pattern and does not contain any additional information
  - Example: UUIDs as names of DCOM objects or classes are pure
- Impure:
  - A name implies additional information about the object (location, function ...)
  - Example: DNS names imply typically additional information, e. g., `login1.cs.hs-rm.de`, pension insurance policy number (contains birth date)

Name Properties

What is a unique name. Give some examples

Name Properties

What is a unique name. Give some examples

Answer

unique: a name identifies uniquely (at most) one object
Examples:

- UUID (Globally Unique Identifier)
- Chassis numbers are unique, licence plates are not (get reused)
- MAC address are theoretically unique, but maybe spoofed and old MAC address ranges get reused
- Electrical or water meter IDs

DNS

Describe the difference between iterative and recursive name resolution.

DNS

Describe the difference between iterative and recursive name resolution.

Answer

- Iterative name resolution
  - The resolution of each partial name starts at the client (beginning at the end)
  - Caching only at the client
- Recursive name resolution
  - Name resolution within the nameserver hierarchy
  - Caching on the servers is possible
  - Requires less communication
  - More load for the root servers

Directory Services

What is a directory service?

Describe the term Distinguished Name (DN) in the context of X.500

Directory Services

What is a directory service?

Answer

- Designed to find addresses (and possibly additional information)
- In comparison to a name service
  - In a directory service entries are typically not looked up primarily by their name, but by their attributes
  - Analogy: „Yellow pages“ vs. phone book
- Common standards
  - X.500
  - LDAP (Lightweight Directory Access Protocol)

Describe the term Distinguished Name (DN) in the context of X.500

Directory Services

What is a directory service?

Answer

- Designed to find addresses (and possibly additional information)
- In comparison to a name service
  - In a directory service entries are typically not looked up primarily by their name, but by their attributes
  - Analogy: „Yellow pages“ vs. phone book
- Common standards
  - X.500
  - LDAP (Lightweight Directory Access Protocol)

Describe the term Distinguished Name (DN) in the context of X.500

Answer

- Each node has a unique name for its layer, called Relative Distinguished Name (RDN)
- Concatenation of RDNs starting at the node up to the root are called Distinguished Name (DN)

Distributed Filesystems

Which consistency semantics for file access are typically used in distributed filesystems?

Why may a newly created file remain invisible for another process in a distributed system for several seconds?

Distributed Filesystems

Which consistency semantics for file access are typically used in distributed filesystems?

Answer

- Strict consistency
- Session semantics
- Read-only files
- Transaction semantics

Why may a newly created file remain invisible for another process in a distributed system for several seconds?

Distributed Filesystems

Which consistency semantics for file access are typically used in distributed filesystems?

Answer

- Strict consistency
- Session semantics
- Read-only files
- Transaction semantics

Why may a newly created file remain invisible for another process in a distributed system for several seconds?

Answer

In NFSv3 directories may be cached and the invalidation happens based on time stamps (which is 30 s for directories)

Storage Systems

Describe difference approaches for storage networks.

Describe the differences between SAN and NAS.

Storage Systems

Describe difference approaches for storage networks.

Answer

- Direct Attached Storage (DAS)
- Storage Area Networks (SAN)
- Network-Attached Storage (NAS)
- Content Addressed Storage (CAS)

Describe the differences between SAN and NAS.

Storage Systems

Describe difference approaches for storage networks.

Answer

- Direct Attached Storage (DAS)
- Storage Area Networks (SAN)
- Network-Attached Storage (NAS)
- Content Addressed Storage (CAS)

Describe the differences between SAN and NAS.

Answer

Essentially: SAN provides block storage, NAS provides network filesystem

Clock Synchronization

Describe the terms accuracy, resolution, stability, offset, and drift with your own words.

Clock Synchronization

Describe the terms accuracy, resolution, stability, offset, and drift with your own words.

Answer

Accuracy How large is the deviation to a reference time

Resolution What is the smallest measurable interval

Offset The time delta to another clock

Drift The frequency delta to another clock

Clock Synchronization (2)

Describe how a computer clock is typically implemented.

Clock Synchronization (2)

Describe how a computer clock is typically implemented.

Answer

- A counter register gets periodically increased or decreased
- A quartz is used as input
- On overflow/underflow the register generates an interrupt
- The counter register loads the value from the reload register

Cristian's Algorithm

Describe the procedure of Cristian's algorithm for clock synchronization.

Cristian's Algorithm

Describe the procedure of Cristian's algorithm for clock synchronization.

Answer

- Passive time server (as reference time source)
- Periodical polling of time by clients
- Measure average round trip time (including processing time on server)
- Set $t_{local} := t_{SRV} + \frac{t_R}{2}$

NTP

Describe the procedure of NTP.

NTP

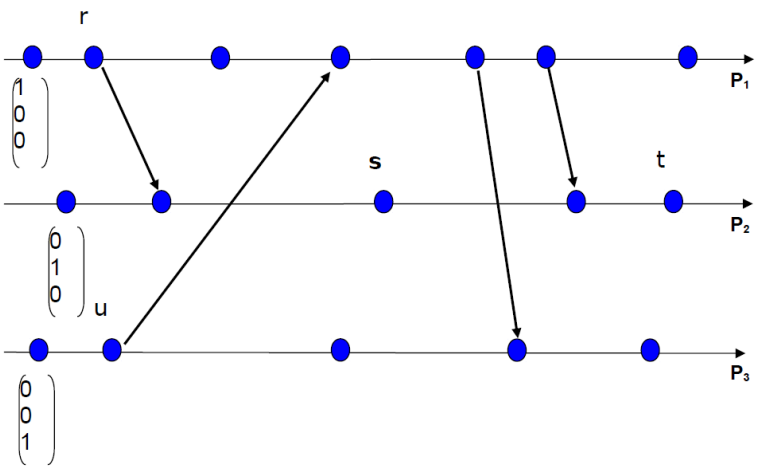
Describe the procedure of NTP.

Answer

- An NTP client polls multiple peers
- The messages contain 64 bit timestamps
- Filters, selection, and clustering algorithms are applied to identify the best time source(s)
- Clock discipline algorithm is used to correct the local clock via adjtime and settimeofday

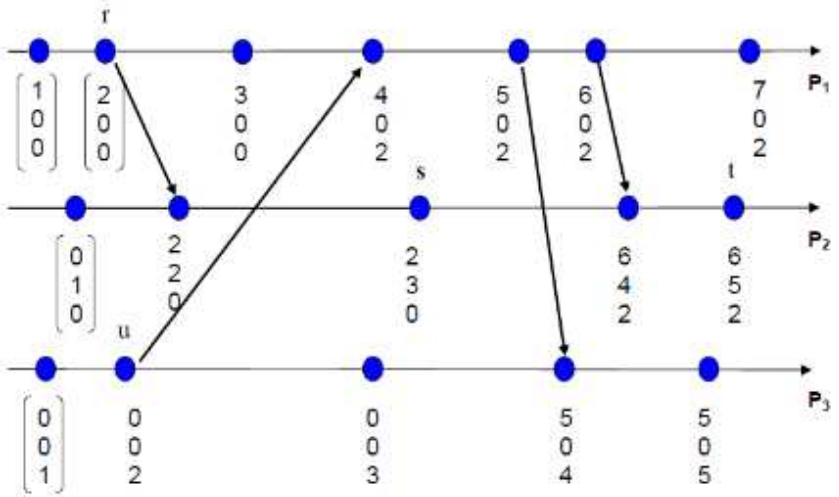
Vector Clocks

Complete the following diagram with the correct vector time stamps.
Between which event pairs of the set r, s, t, u a causal dependency exists.
Explain!



Vector Clocks

Complete the following diagram with the correct vector time stamps.
Between which event pairs of the set r, s, t, u a causal dependency exists.
Explain!



Answer

$(r, s) because VC(r) < VC(s)$
$(s, t) because in the same process => (r, t)$
$(u, t) because VC(u) < VC(t)$
$r||u, s||u,$

Security

Describe the difference between authentication and authorization

Describe the difference between symmetrical and asymmetrical encryption methods

Security

Describe the difference between authentication and authorization

Answer

- Authentication:
  - Validation of an identity
  - Mutual authentication of communication peers may be required
- Authorisation:
  - Grant permissions

Describe the difference between symmetrical and asymmetrical encryption methods

Security

Describe the difference between authentication and authorization

Answer

- Authentication:
  - Validation of an identity
  - Mutual authentication of communication peers may be required
- Authorisation:
  - Grant permissions

Describe the difference between symmetrical and asymmetrical encryption methods

Answer

- Symmetrical Encryption
  - a secret key for encryption and decryption
  - requires a secure channel for key distribution
- Asymmetrical Encryption
  - A pair of keys is required (private and public key)

Cryptographic Hash Functions

Describe the principle of cryptographic hash functions

Cryptographic Hash Functions

Describe the principle of cryptographic hash functions

Answer

- Generation of a digital fingerprint of documents or messages, called message digest
- Base for digital signatures
- Hash function H
  - $h = H(P)$
  - Message P of arbitrary length
  - h bit sequence of fixed length
  - Comparable to CRC for error detection
- Assumptions
  - Calculation of H is simple
  - Reversal, i. e., identifying the original message for a given hash value is difficult (one-way function)
  - Modification of document P results in a different hash value h

Digital Signatures

Describe the principle of digital signatures

Describe the functioning of SSL/TLS.

Digital Signatures

Describe the principle of digital signatures

Answer

- Sign by the encryption of the hash value of a message using the private key
- Public key is used on the receiver side to verify the signature

Describe the functioning of SSL/TLS.

Digital Signatures

Describe the principle of digital signatures

Answer

- Sign by the encryption of the hash value of a message using the private key
- Public key is used on the receiver side to verify the signature

Describe the functioning of SSL/TLS.

Answer

- Uses a PKI
- Based on certificates (containing the public key of the certificate owner and their name)
- Certificates are signed certificate authorities (CAs)
- CAs form a chain of trust up to the root CAs