End-to-End Arguments in System Design

J.H Saltzer, D.P. Reed, and D.D Clark, 1984

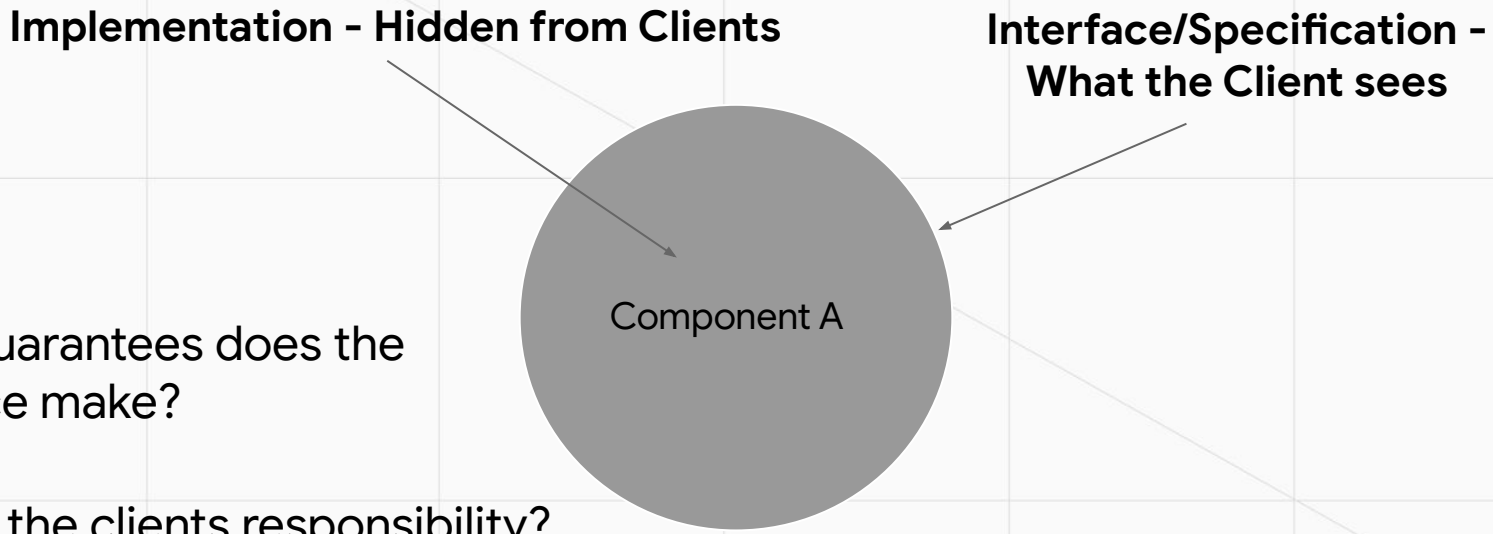
Presented by Tedi Mitiku in CS 6410 on Sept. 22nd, 2022

Agenda

- Motivation
- End-To-End Principle
- File Transfer Example
- Applying the End-To-End Principle
- Performance Considerations
- Discussions

Motivation

Recap: Interface/Implementation



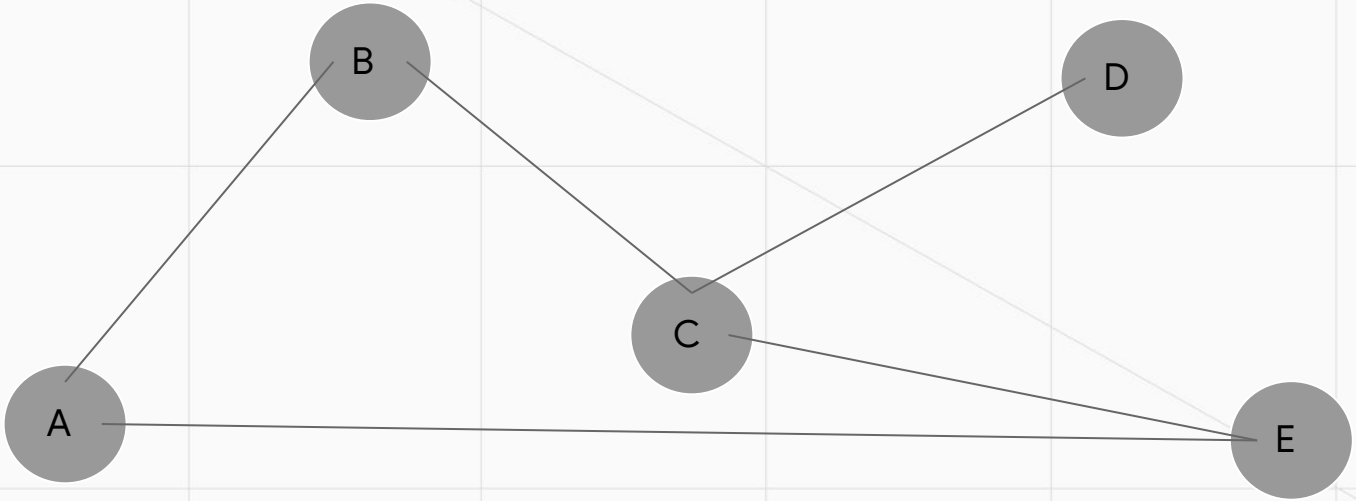
What guarantees does the interface make?

What is the clients responsibility?

What functions does the interface provide?

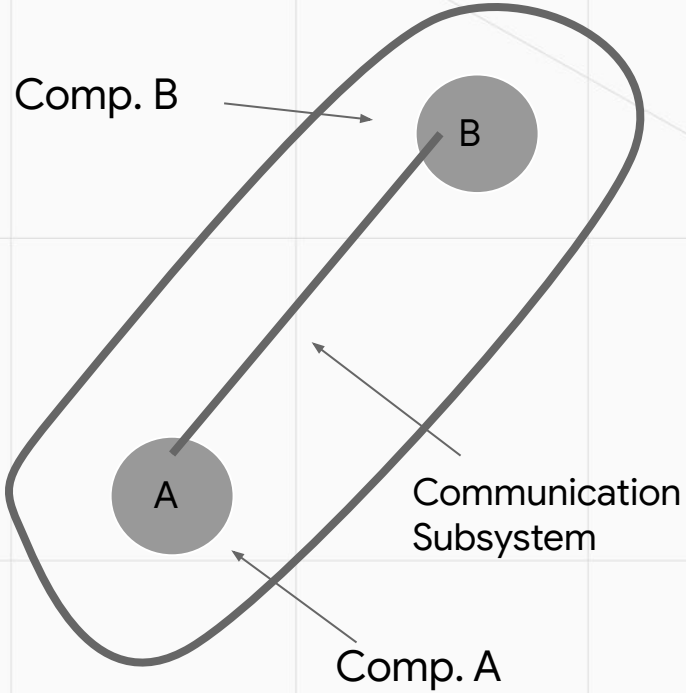
Motivation

Interface/Implementation for Multi-Component Systems



Motivation

Component Design Considerations



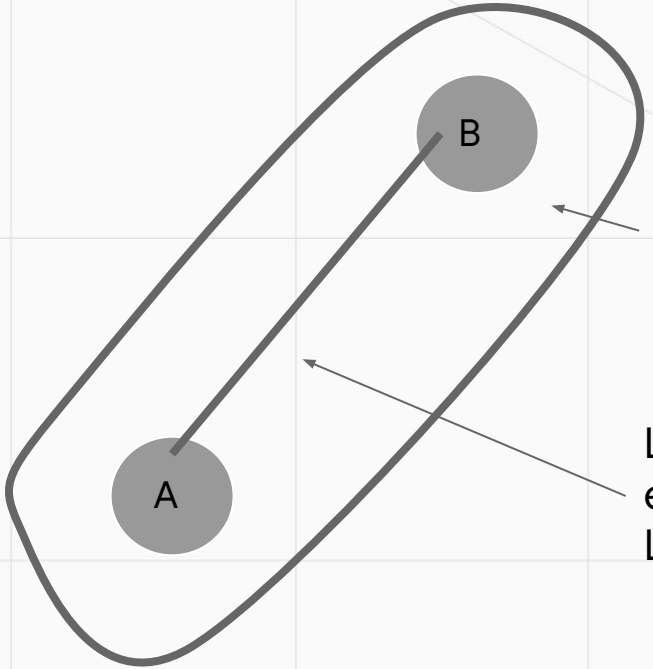
What's the responsibility of each component? (interface)

What components should be responsible for certain functions/guarantees?

Which component is being designed first?

Motivation

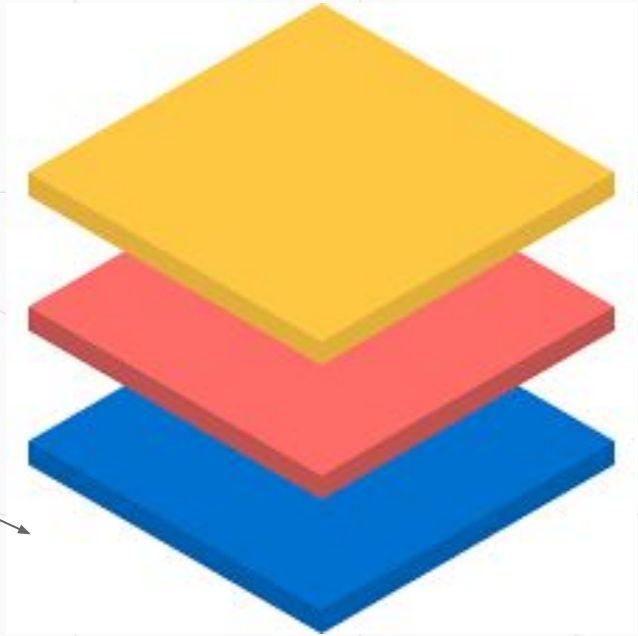
End-to-End Intuition



High Level (End)
eg. Application
Layer

Low Level
eg. Data Link
Layer

Layers of Abstraction

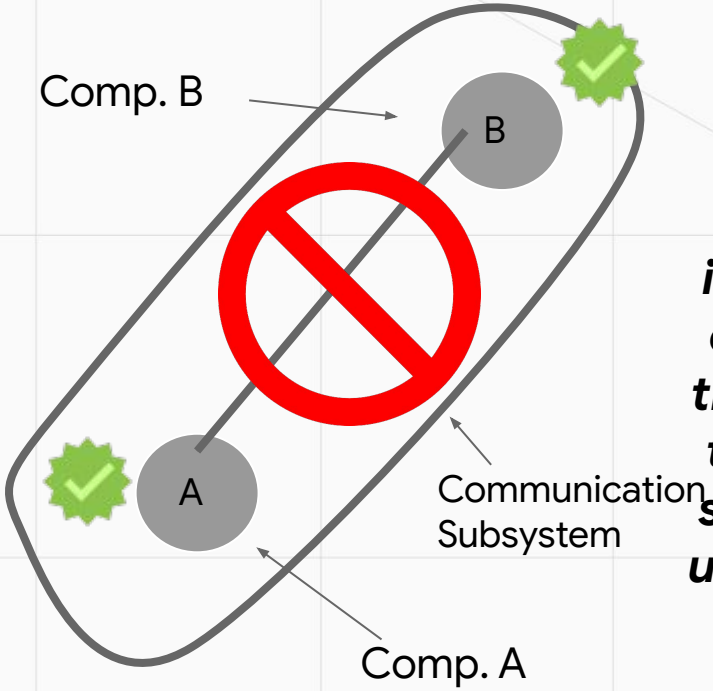


Agenda

- Motivation
- End-To-End Principle
- File Transfer Example
- Applying the End-To-End Principle
- Performance Considerations
- Discussions

End-to-End Principle

End-to-End Principle

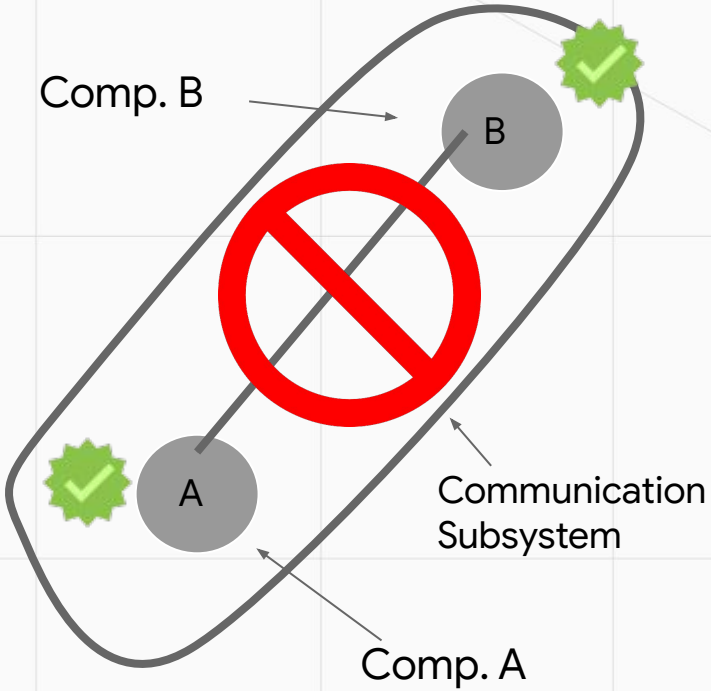


In this scenario, where should a function X be implemented?

"The function can completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system. Therefore, providing the function as a feature of the communication system itself is not possible. (Although could be useful in cases as a performance enhancement)"

End-to-End Principle

End-to-End Principle (Simpler Terms)



In this scenario, where should a function X be implemented?

1. To implement function X correctly, you need knowledge of the application.

2. Communication system has no knowledge of application.

3. Therefore, not possible to implement function X in communication system.

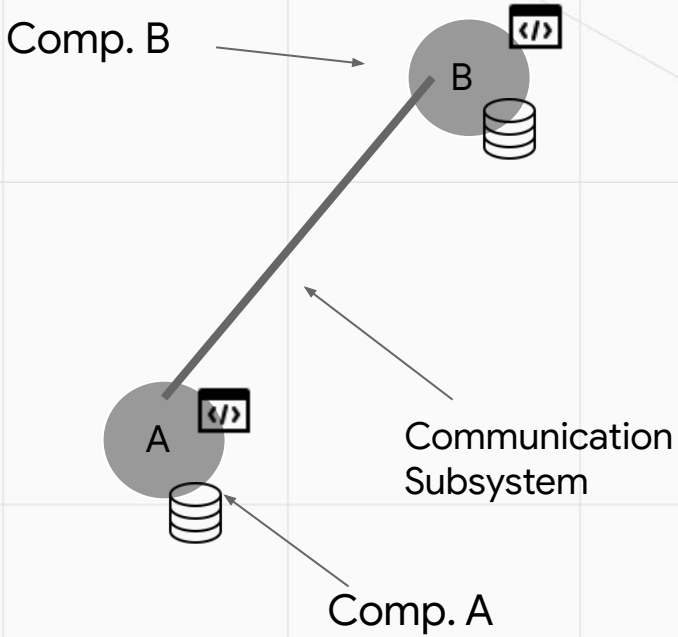
*Except in certain cases for performance improvements

Agenda

- Motivation
- End-To-End Principle
- File Transfer Example
- Applying the End-To-End Principle
- Performance Considerations
- Discussions

File Transfer Example

Setup

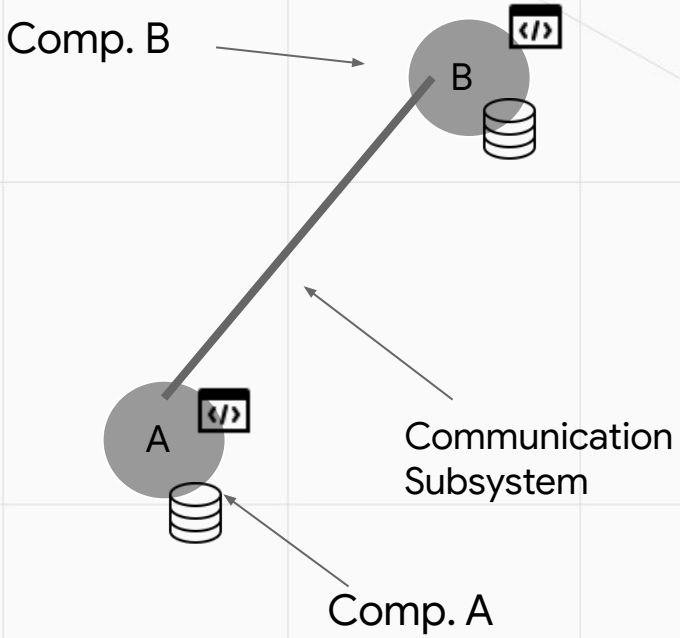


Components:

- Component's A and B: Host Machines with
  - File Transfer App
  - File System
  - Datastore
- Communication Subsystem: Data communication network

File Transfer Example

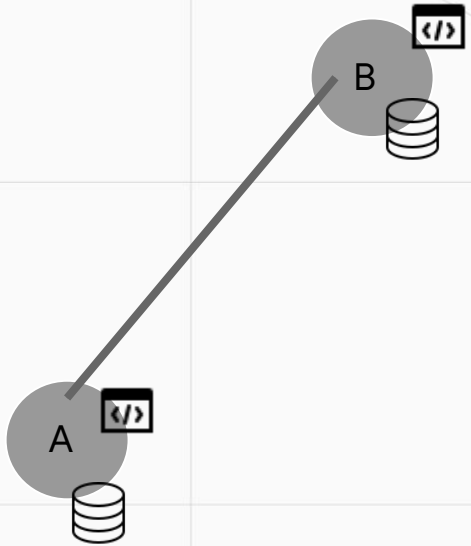
Steps



1. File transfer app **A** reads files from file system
2. File transfer app **A** uses **data comm network** to send packets with data
3. **Data comm network** moves packets from **A** to **B**
4. **Data comm network** retrieves packets for file transfer app **B**
5. File transfer app **B** asks file system to write data

File Transfer Example

Concerns

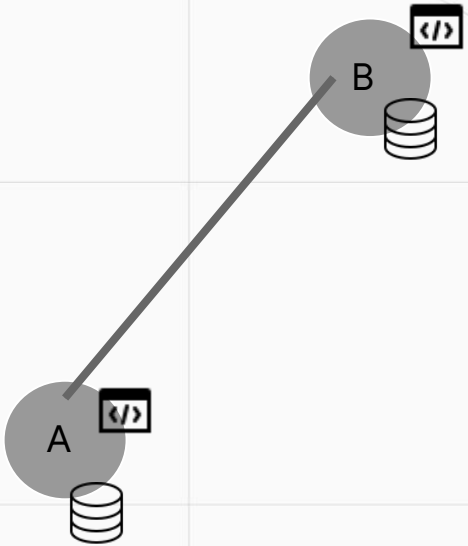


Concern 1:

File sent to B contains incorrect data due to hardware fault in disk

File Transfer Example

Concerns

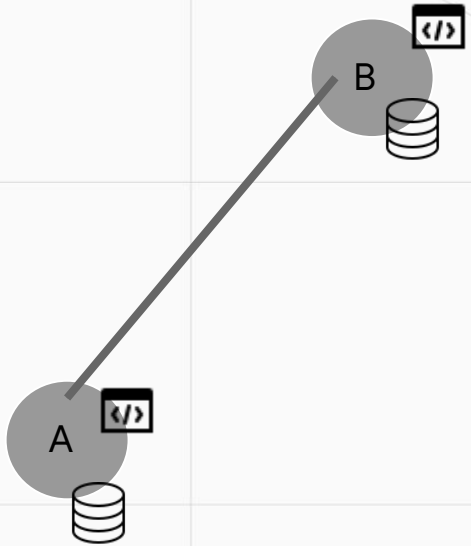


Concern 2:

Bug in file transfer app causes data to be processed wrong

File Transfer Example

Concerns

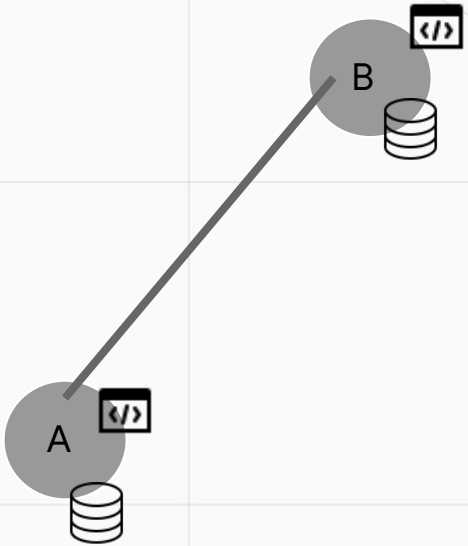


Concern 3:

Hardware processor or local memory error causes data to be corrupted

File Transfer Example

Concerns

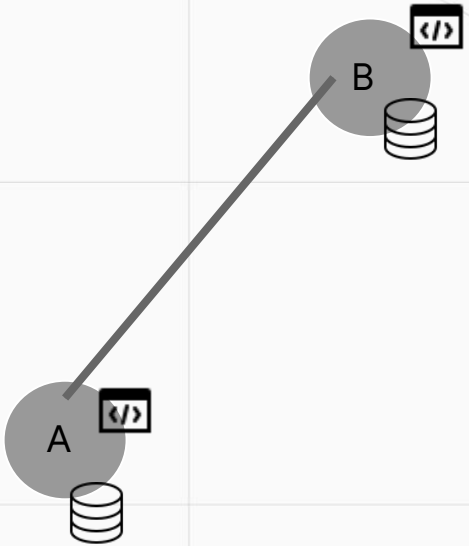


Concern 4:

Data comm network drops bits in packets

File Transfer Example

Concerns

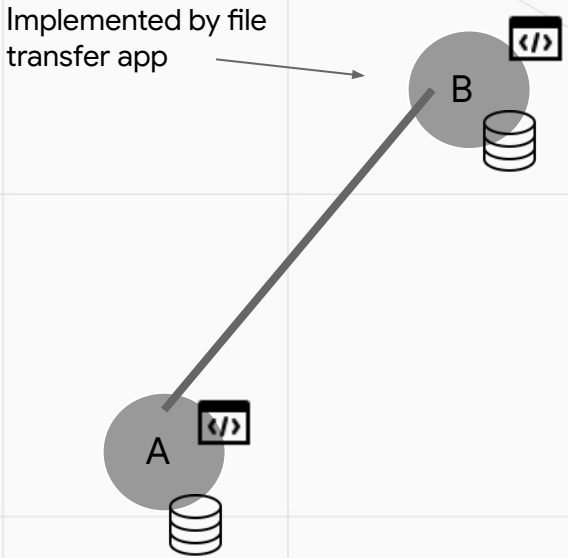


Concern 5:

Host (A or B) crashes.

File Transfer Example

Potential Solutions



Triple check everything

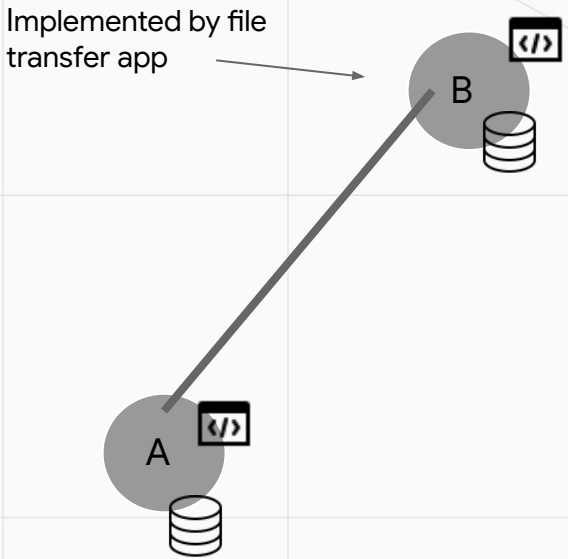
eg. duplicate copies, timeout, retry, redundant error checking

But...

Inefficient.

File Transfer Example

Potential Solutions



End-to-end check and retry

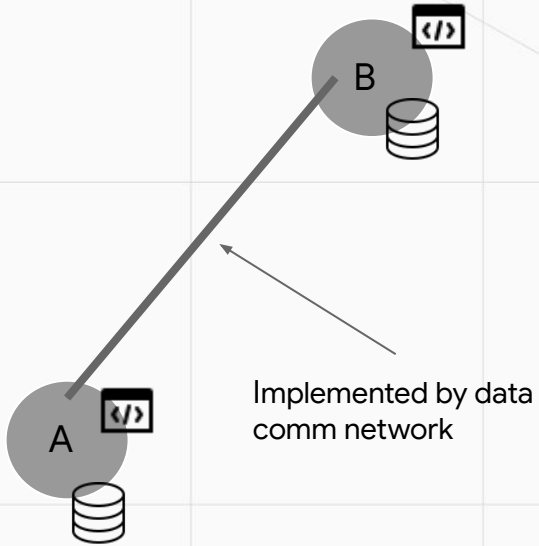
eg. add a checksum step

But...

How many retries do you do?

File Transfer Example

Potential Solutions



Guarantee of reliable data transmission

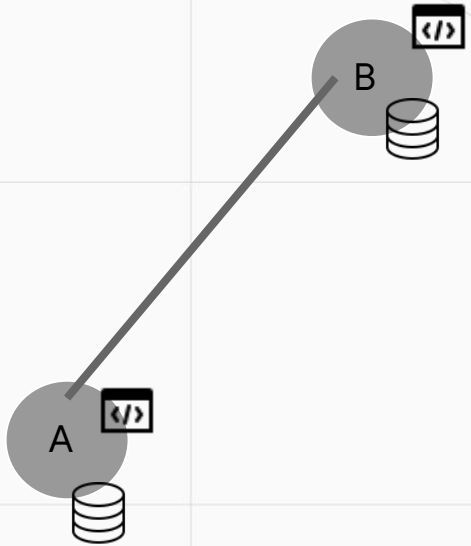
eg. selective redundancy

But...

Does this really help that much?

File Transfer Example

Applying End-to-End Principle

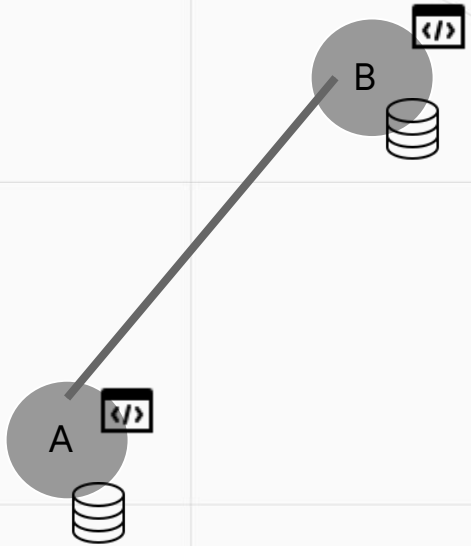


1. To implement **careful file transfer** correctly, you need knowledge of the application.

Addressing concerns 1, 2, 3, and 5 requires knowledge of application.

File Transfer Example

Applying End-to-End Principle



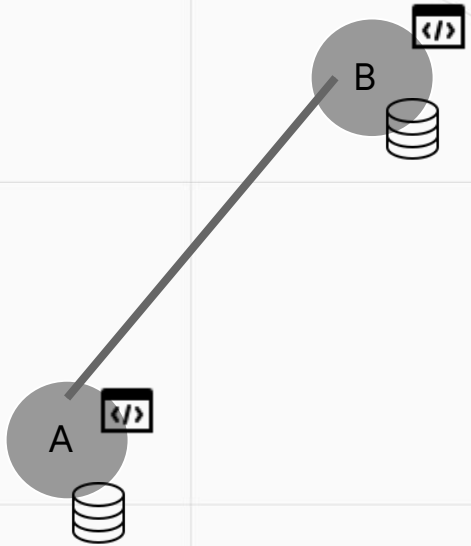
1. To implement **careful file transfer** correctly, you need knowledge of the application.

2. Communication system has no knowledge of application.

3. Therefore, need not implement function in communication system for correctness.

File Transfer Example

So what? Why not do it anyways?



- Other applications that utilize comm system now locked into all functions of comm system

- More functions in comm system -> redundant checks -> decreased performance

Agenda

Motivation

End-To-End Principle

File Transfer Example

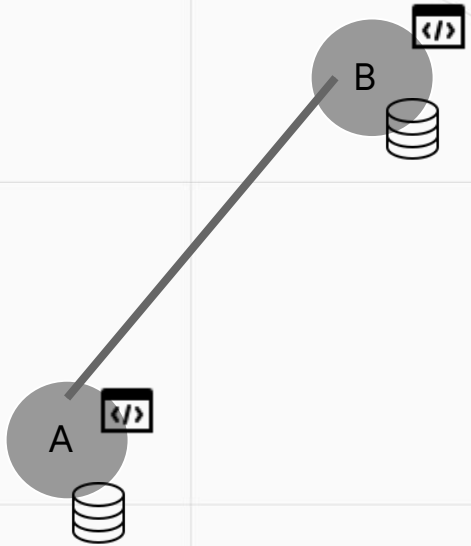
Applying the End-To-End Principle

Performance Considerations

Discussions

Applying End-to-End Principle

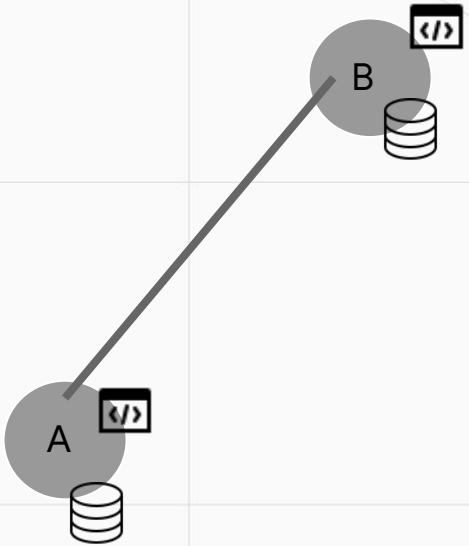
Delivery Guarantees



Function X: Acknowledgement of Delivery

Applying End-to-End Principle

Delivery Guarantees



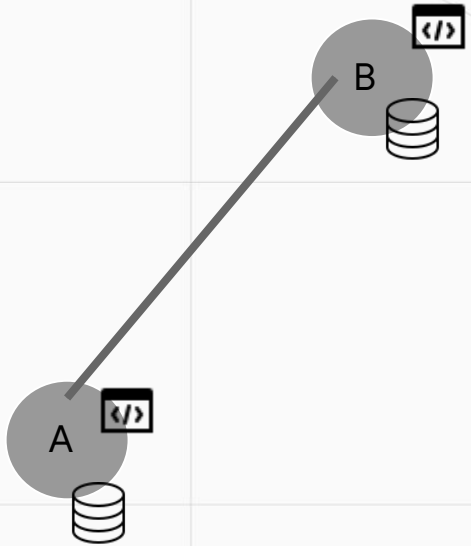
Function X: Acknowledgement of Delivery

1. To implement **function X** correctly, you need knowledge of the application.

Yes. Application wants to know if target host ACTED on message (not just received it).

Applying End-to-End Principle

Delivery Guarantees



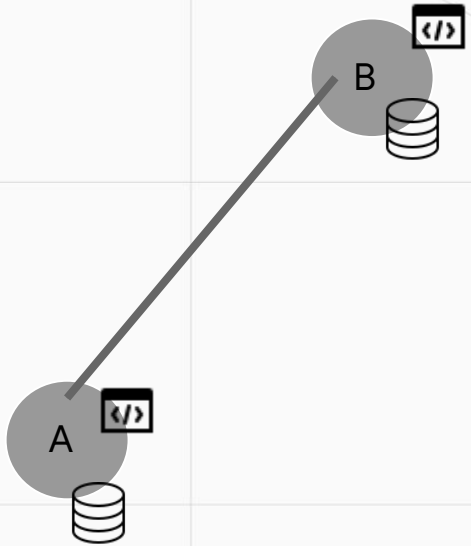
1. To implement **acknowledgement of delivery** correctly, you need knowledge of the application.

2. Communication system has no knowledge of application.

3. Therefore, need not implement **acknowledgement of delivery** in communication system for correctness.

Applying End-to-End Principle

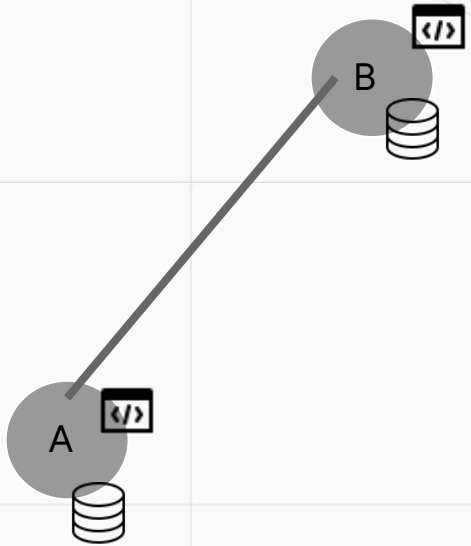
Secure Transmission of Data



Function X: Data Encryption

Applying End-to-End Principle

Secure Transmission of Data



Function X: Data Encryption

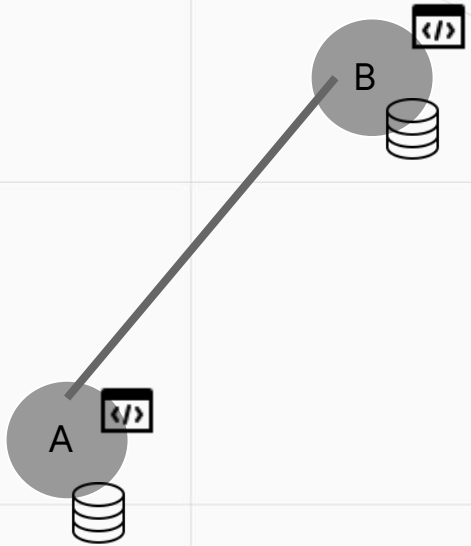
1. To implement **function X** correctly, you need knowledge of the application.

Yes. Application needs to check authenticity of message anyways.

Also, providing e2e encryption prevents data being exposed to public.

Applying End-to-End Principle

Secure Transmission of Data



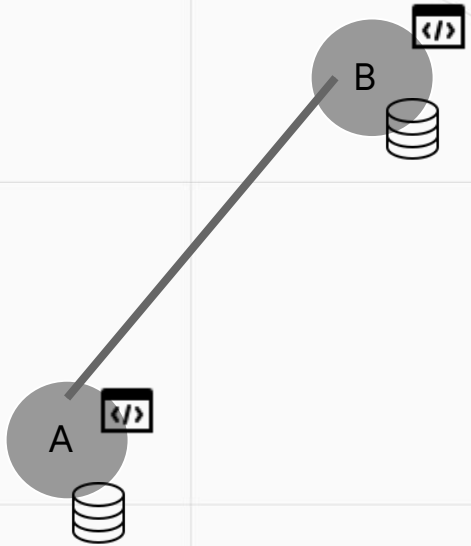
1. To implement **data encryption** correctly, you need knowledge of the application.

2. Communication system has no knowledge of application.

3. Therefore, need not implement **data encryption** in communication system for correctness.

Applying End-to-End Principle

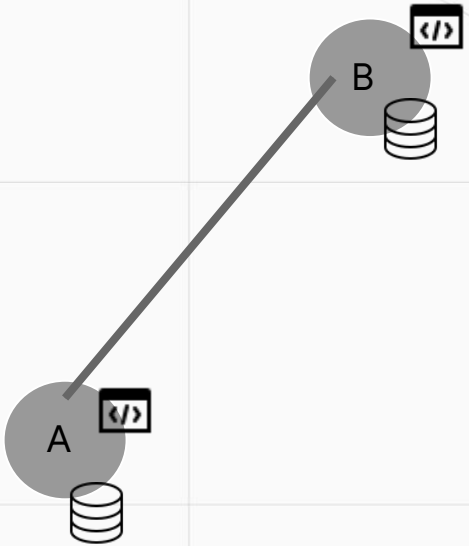
Duplicate Message Suppression



Function X: Duplicate Message Suppression

Applying End-to-End Principle

Duplicate Message Suppression



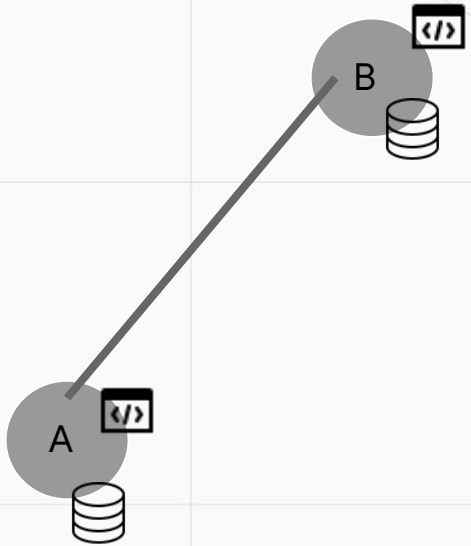
Function X: Duplicate Message Suppression

1. To implement **function X** correctly, you need knowledge of the application.

Yes. Only application knows how to detect and handle application level duplications.

Applying End-to-End Principle

Duplicate Message Suppression



1. To implement **duplicate message suppression** correctly, you need knowledge of the application.

2. Communication system has no knowledge of application.

3. Therefore, need not implement **duplicate message suppression** in communication system for correctness.

Agenda

Motivation

End-To-End Principle

File Transfer Example

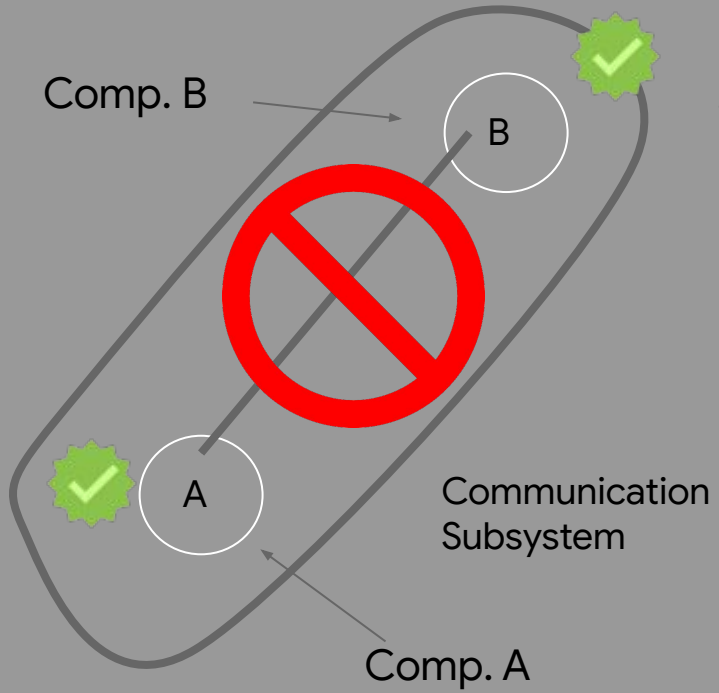
Applying the End-To-End Principle

Performance Considerations

Discussions

End-to-End Principle

End-to-End Principle (Simpler Terms)



In this scenario, where should a function X be implemented?

1. To implement function X correctly, you need knowledge of the application.

2. Communication system has no knowledge of application.

3. Therefore, should not implement function in communication system.

*Except in certain cases for performance improvements

Performance Considerations

Trade Offs (File Transfer Example)

- Performance decreases as size of checksum or distance increases
- Higher possibility of packet loss

Performance Considerations

Trade Offs (File Transfer Example)

No Reliability — **Guaranteed Reliability**

- Lots of redundant checks -> decreases performance
- Other applications locked into functions they don't need
- Low level == less knowledge

Performance Considerations

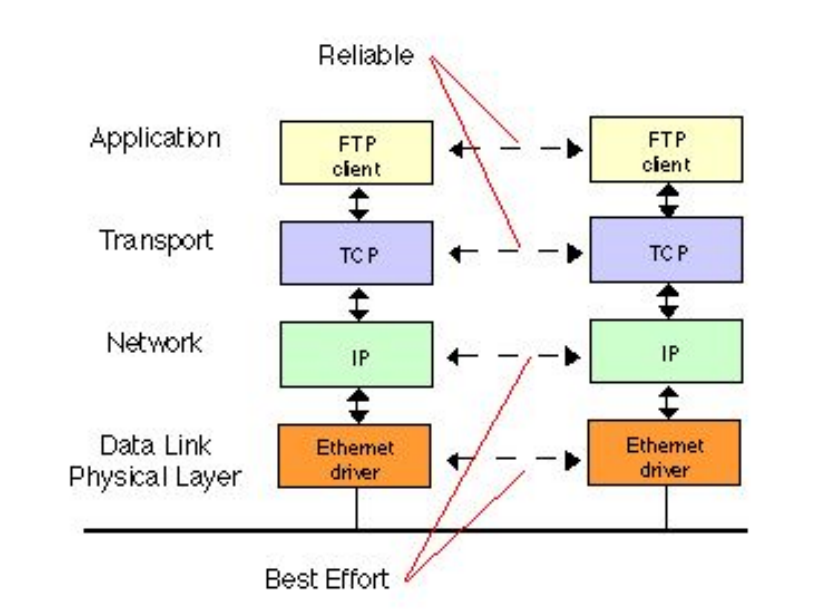
Insights

1. End-to-end principle is a guideline not **a rule.**

2. Provide “low-level” functionality as a **performance improvement**, not a correctness requirement.

Performance Considerations

Example: Transmission Control Protocol (TCP)



- **Offers:**
  - Ordered data transfer
  - Retransmission of lost packets
  - Error-free data transfer
  - Flow control
  - Congestion control

- **Insight:** Can provide some guarantees for performance, even if not enough.

Agenda

Motivation

End-To-End Principle

File Transfer Example

Applying the End-To-End Principle

Performance Considerations

Discussions

Discussions

1. How do we identify the “ends”?

2. What are examples of applications where functionality in communication system provides performance enhancements?

3. What are examples of applications where functionality in communication system is required for correctness?

4. Practice: Apply End-to-End Principle to
   a. Guaranteed FIFO message delivery
   b. Transaction management