The Internet Today

Niko Matsakis

Outline

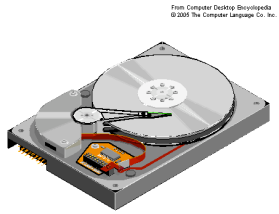
- Summaries of:
  - End-to-End Arguments in System Design
  - The Design Principles of the DARPA Internet Protocols
- Criticisms and Commentary
- Conclusion

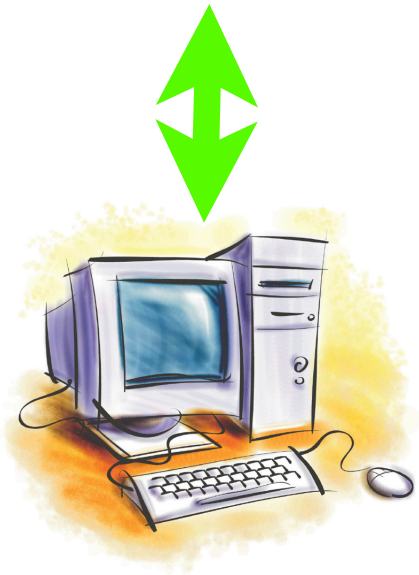
The End-to-End Argument

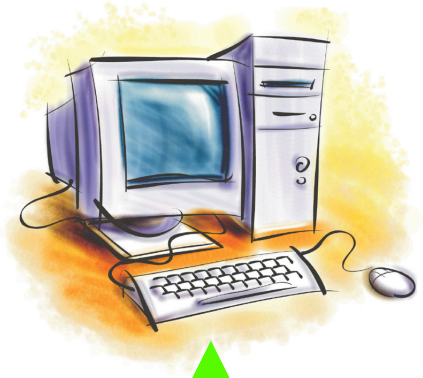
- E2E founded on the observation that:
  - every application has different needs.
- The argument:
  - There is no one-size-fits-all “solution."
  - Therefore, move functionality as close to the application as possible.

Careful File Transfer

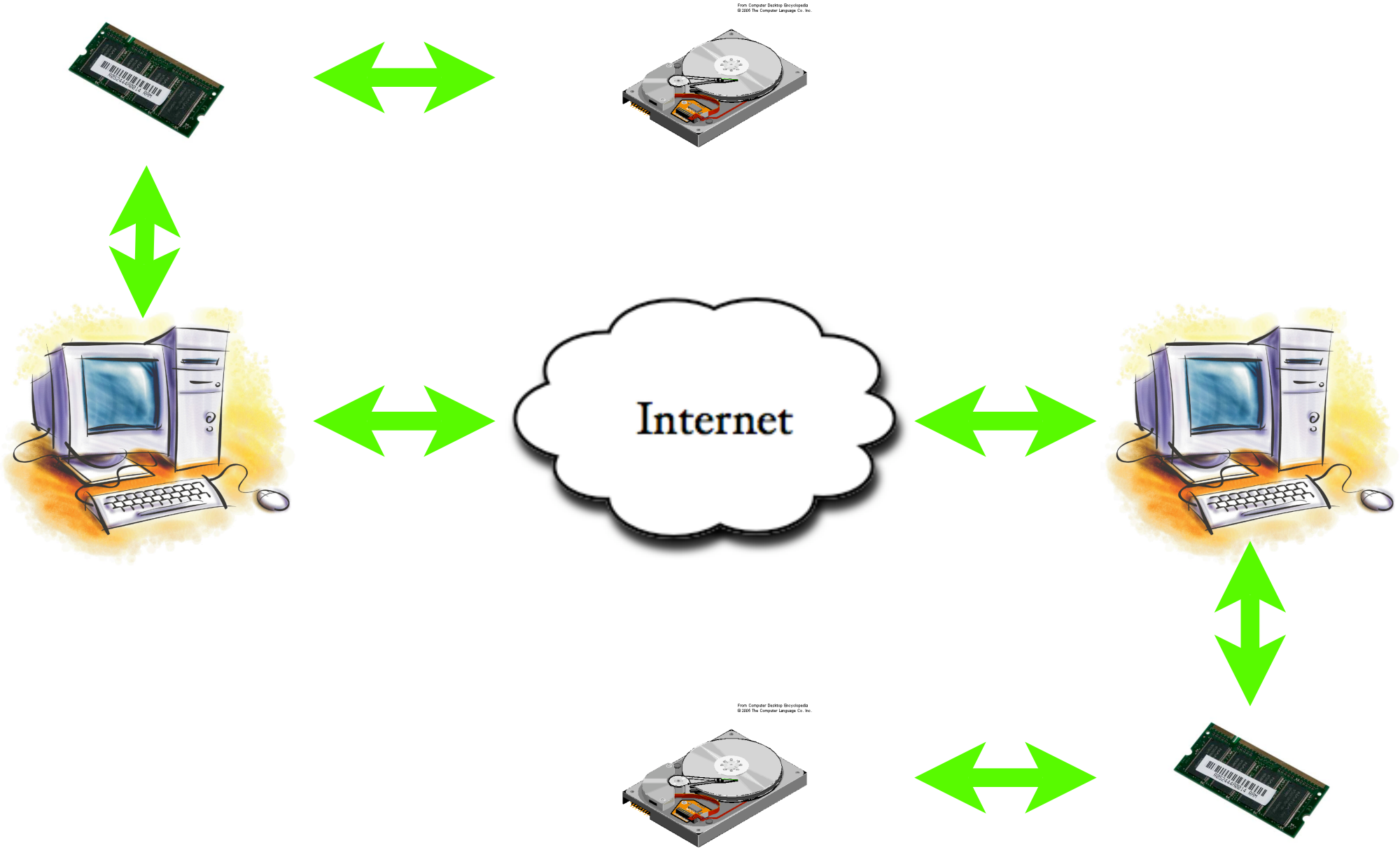
Careful File Transfer







Careful File Transfer



End-to-End Solution

- Store a checksum on the disk
- Destination reads what it wrote back from the disk to compare the checksum
- One check suffices to detect all possible sources of error
  - besides an incorrectly coded checksum routine...

Think it can't happen?

- Included in the paper is an example from MIT, where a hardware failure caused occasional corruption of packets en route.

Performance Considerations

- Lower levels may play a role in providing higher functionality for performance reasons
- Must be careful to avoid taxing all users of the lower level with a feature that supports only one application

Other Examples

- End-to-end applies in many other scenarios.

Delivery Guarantees

- Suppose I am ordering something over the Internet. How do I know my order was received?

Delivery Guarantees

- One solution: the Internet tells you when your packets arrive.
  - Is that enough?

Delivery Guarantees

- One solution: the Internet tells you when your packets arrive.
  - Is that enough?

Delivery Guarantees

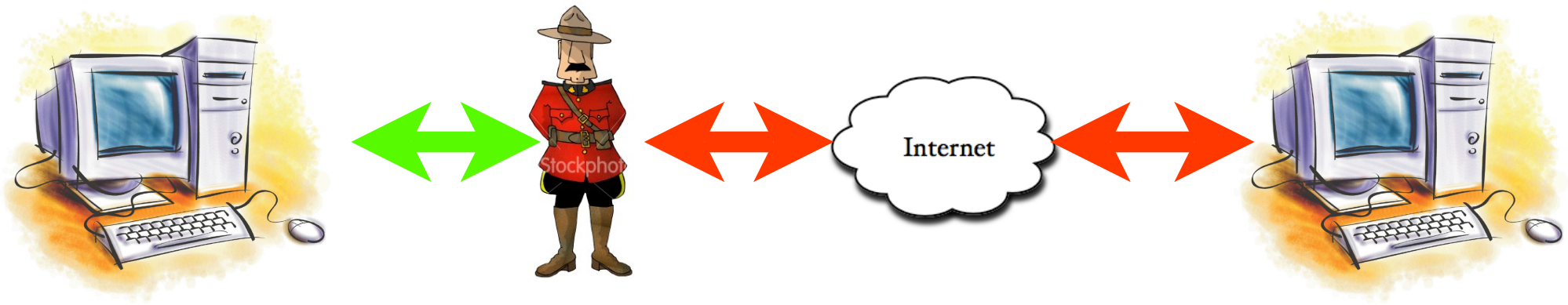
- Better solution: eBay tells you when your order is complete.

Encryption

- Problem: my purchase is in the clear, and I don't know who user "isell2you" is anyway

Encryption

- One solution: Introduce an intermediary.
  - Key distribution?
  - Still some distance in the clear?
  - Authentication?



Encryption

- Better solution: encrypt it myself!

Beyond Correctness

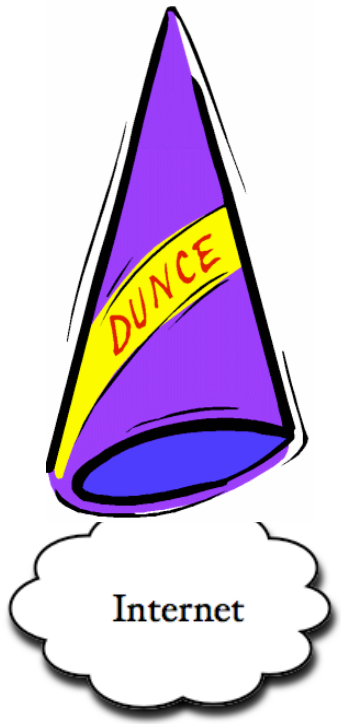
- End-to-end offers other benefits:
  - No need to change infrastructure to deploy a new service
    - Immediate benefits
  - Decentralized control
  - Simpler, more reliable internal network

Identifying the End Points

- Identifying the end points can be subtle:
  - Telephone conversation: human
  - Message recorder: answering machine
  - Different tradeoff for delay versus accuracy

Conclusions

- Applying the E2E principle results in:
  - a system where each layer provides *only the minimum functionality required by all applications*
- So-called "stupid network"
- Benefits:
  - correctness and flexibility



Design Philosophy

- "Design Philosophy of the DARPA Internet"
- Explain the reasoning that led to the current structure of the Internet.

Etymology

inter • net

- For many, the words *internet* and *computer network* are synonymous.

Etymology

inter • net

- The primary purpose of the internet, however, was to *interconnect* existing networks.
  - ARPANET, ARPA Radio Network, etc.

Guiding Goals

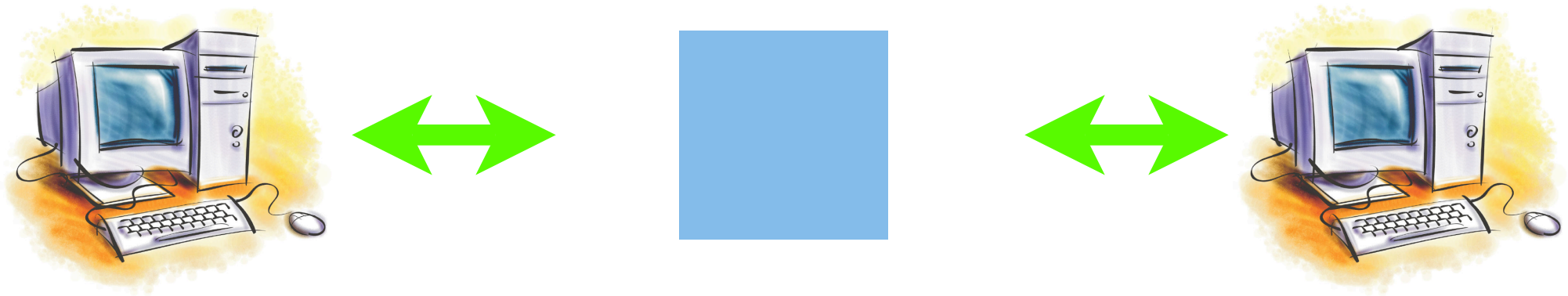
- The paper identifies 7 design goals overall. Here are the 3 most important:
  - **Resiliency**: Network must operate even when intermediate nodes fail
  - **Service flexibility**: Multiple types of services must be supported
  - **Network flexibility**: Must accomodate a variety of networks

Fundamental Design

- The 3 primary goals led directly to the fundamental design of the internet as a *datagram* service.
- Primary function of the network:
  - Best effort delivery of small packets
- The "smarts" are in the end nodes
  - End-to-end principle at work

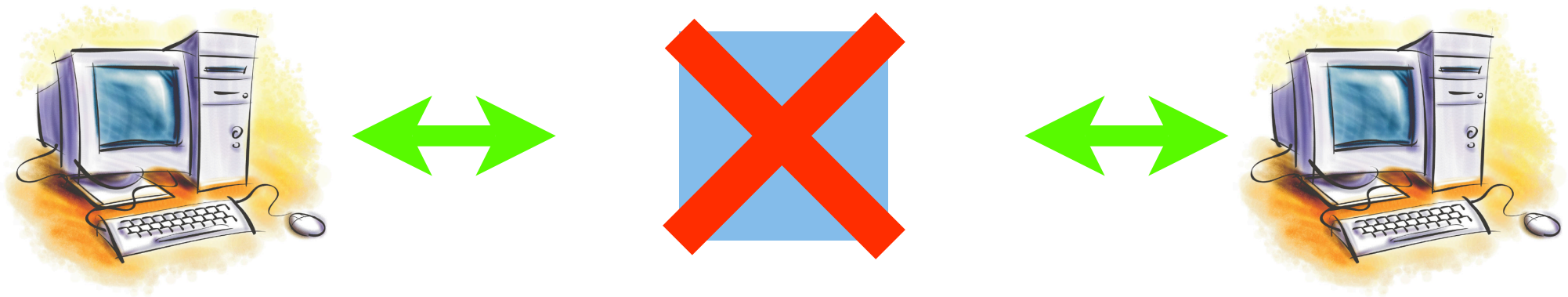
Resiliency

- A conversation consists of a large set of intermediate state
- If an intermediary dies, this state must be preserved for the conversation to continue



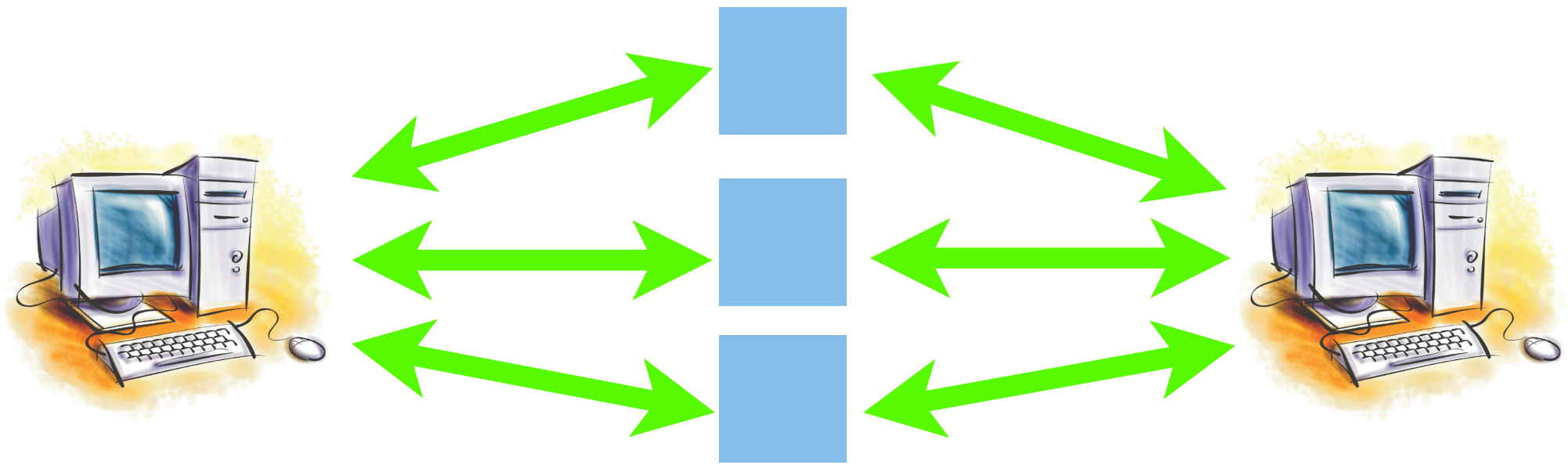
Resiliency

- A conversation consists of a large set of intermediate state
- If an intermediary dies, this state must be preserved for the conversation to continue



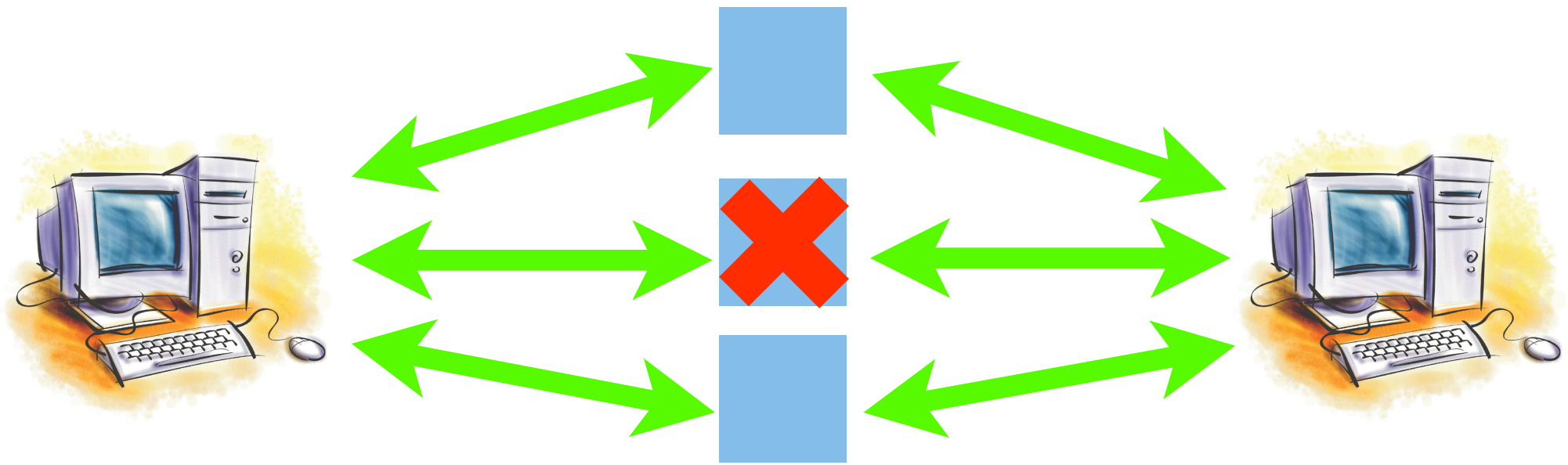
Resiliency

- One solution: Reproduce this state information across intermediaries.
  - Complex
  - Can only cope with *k* failures



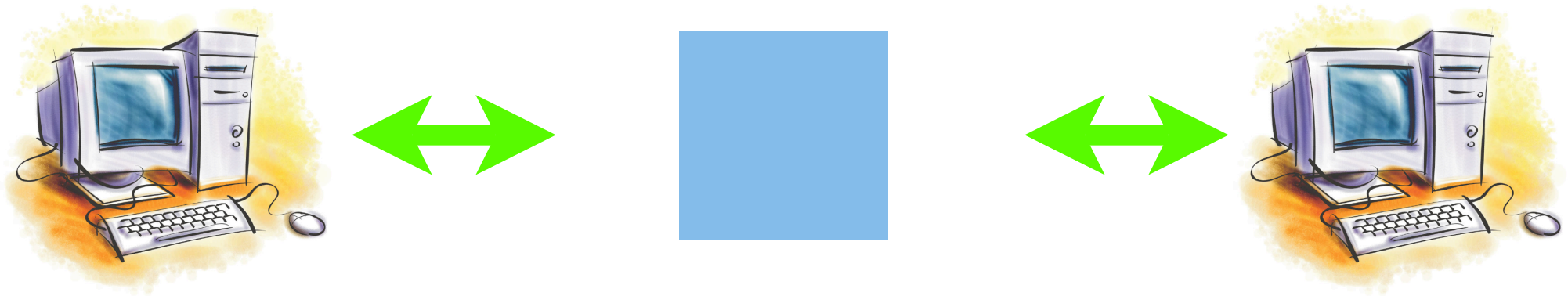
Resiliency

- One solution: Reproduce this state information across intermediaries.
  - Complex
  - Can only cope with *k* failures



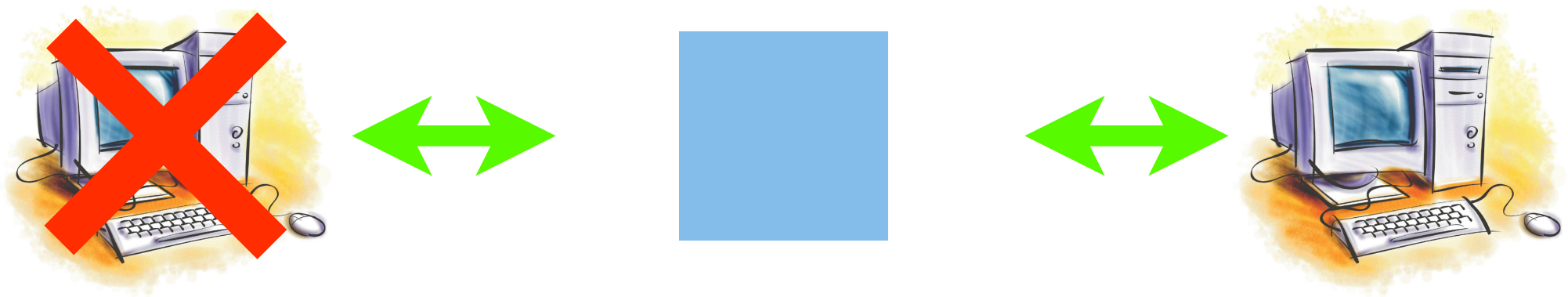
Resiliency

- Better solution: Fate-sharing
  - End node *itself* stores the state
  - Intermediaries know nothing



Resiliency

- Better solution: Fate-sharing
  - End node *itself* stores the state
  - Intermediaries know nothing



Service Flexibility

- TCP was initially thought to be enough.
- Several services not well supported:
  - Live Voice communication
  - Long-distance debugging

Service Flexibility

- Datagrams allow each service to customize the reliability/latency tradeoff
- Few services are built on datagrams directly; they serve as a building block.

Network Flexibility

- Datagrams can be supported by a variety of networks
- Because complex protocols, such as TCP, are regulated by the end nodes, they can operate over any network

Successes of the Internet

- Barely need mentioning
- Dizzying array of applications
- All manner of networks --- from telephony to fiber-optics --- have successfully been integrated

Downsides of the Datagram

- The design of the Internet as a datagram service has downsides as well:
  - Inefficiency
  - Abuse and poor implementation
  - Lack of accountability
- All relate to the ignorance of intermediate nodes

Inefficiency

- Intermediate nodes cannot assist in communication except in the simplest way.
- For example, retransmitted packets must must cross the entire internet again.

Abuse and Implementation

- Intermediaries cannot police the net.
- End nodes responsible for congestion.
- Poor implementation or intentional abuse can harm network performance for everyone.

Accountability

- Most communications take place in sequence, not isolated datagrams.
- Routers and gateways are ignorant of these communications, making accountability very difficult.

Criticisms

- Recently there have been many assaults on the end-to-end principle:
  - Political
  - Technical

Political

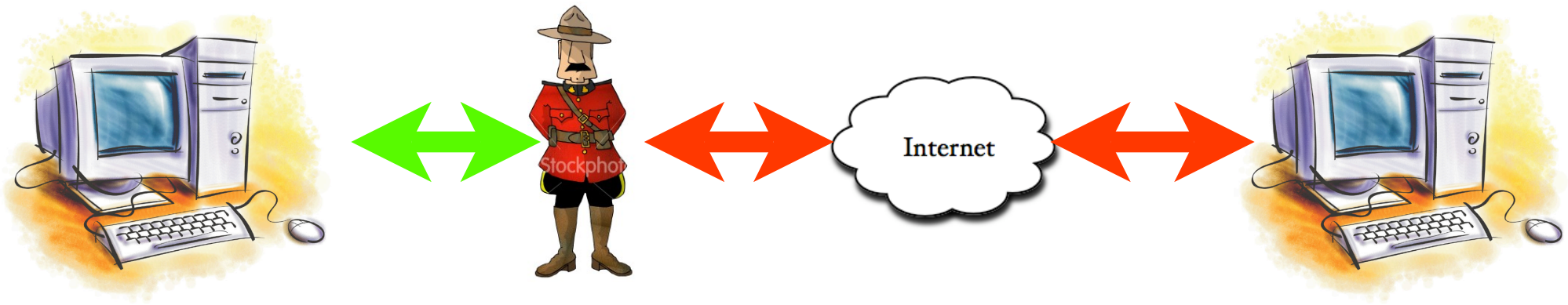
- ISP Differentiation
  - ISP provides the network, which is a commodity. Where is the money?
    - Network Neutrality
- Governmental and corporate agents
  - Taxation, censorship, enforcement of laws and regulations.

Technical

- Trust:
  - Spam, DOS, and other malevolent end-user behavior
- Streaming Content, Quality of Service:
  - IP treats all packets alike
- Caching:
  - 2-tier structure

Trust and Naïve Users

- Recall the rejected model for encryption
- Effectively, this is a firewall or filter
- Given naïve or untrusted users, such a model may in fact be necessary



End-to-End?

- In fact, many standard network devices are not entirely consistent with E2E:
  - Firewalls and filters
  - Network Address Translation (NAT)
  - Content-based Routing
  - http://anonymizer.com

N-A-T

192.67.0.2

18.224.0.56

22.1.0.3

192.67.0.1

N-A-T

- What must be updated?
  - IP Headers
  - TCP Headers
  - Any protocol headers which mention the translated IP address

Lack of Information

- E2E assumes that the end node has more knowledge than the intermediaries
- But not always the case:
  - congestion, routing, trust
- Even for reliability, the prevalence of TCP indicates a need for a reliable communication primitive

Conclusions

- End-to-end principle has been and remains very important to the internet.
- Some things, however, may be best addressed in the network itself.
- Not only a technical question, but also a legal, ethical, and political one.

References

- "Active Networking and End-to-End Arguments", D. Reed, J. Salzer, D. Clark
- "Rethinking the design of the internet: The end to end arguments vs the brave new world", D. Clark and M. Blumenthal
- RFC 3724: The rise of the Middle and the Future of End-to-End