Building Trust in Distributed Systems using Trusted Execution Environment

April 26, 2023

Ines Messadi and Rüdiger Kapitza

Friedrich-Alexander-Universität Erlangen-Nürnberg

Lehrstuhl für Verteilte Systeme
und Betriebssysteme

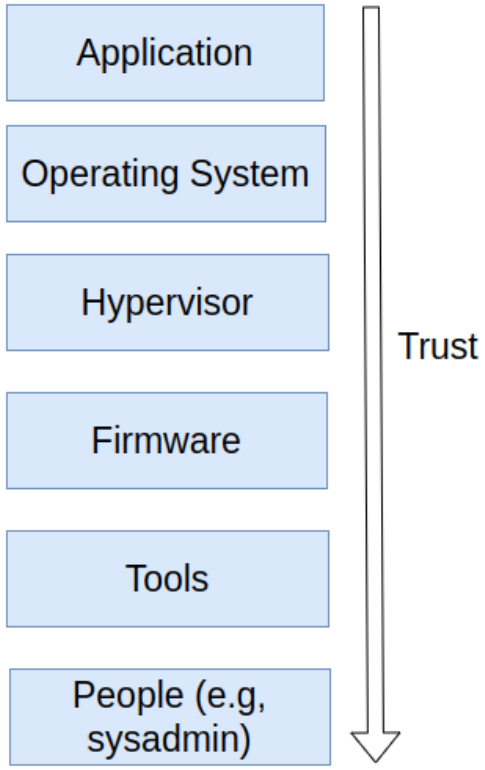
FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Background

How much to trust?

- Huge trusted computing base
  - Privileged software, hypervisor, firmware
  - Staff working on the machines
  - System administrators



Confidential Computing Idea

We don't trust the host at all $\rightarrow$ it is assumed *malicious*
Data protection is important at runtime

Use-case: Running software securely

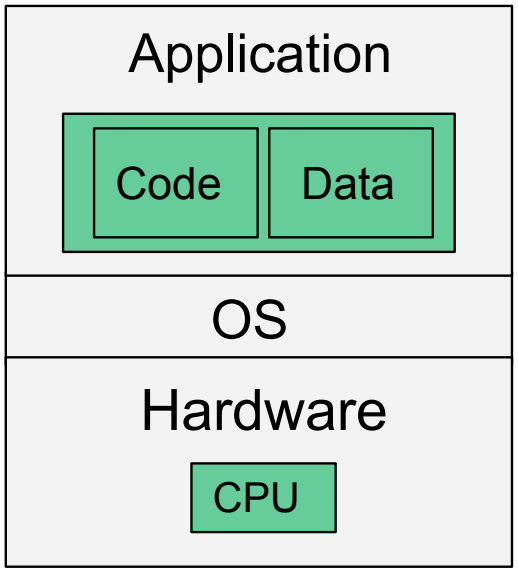
- I want to run my software somewhere but securely
- I want to have the proof that it is running securely
- I don't trust the remote system
  - Computation without data disclosure
    - SSH keys, encryption keys, medical data

$\rightarrow$ data is not leaked

$\rightarrow$ the provider or administrator cannot see what I do

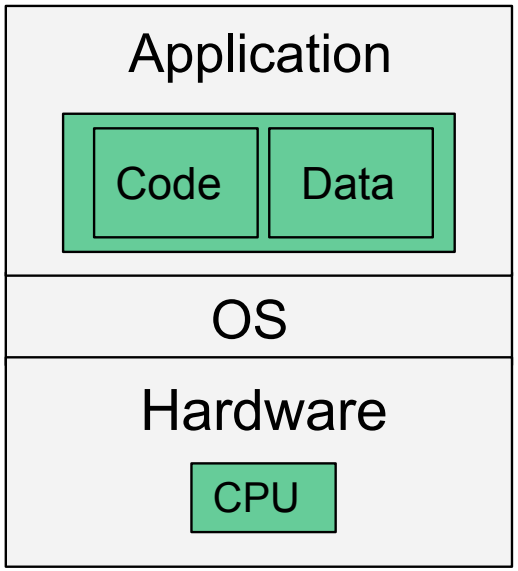
Trusted Execution Environment (TEE)

- Securely running sensitive application
  - **Isolated, always** encrypted enclaves
- Administrator
  - **Cannot** influence or monitor the application
- Authentication, Integrity, Confidentiality, Privacy



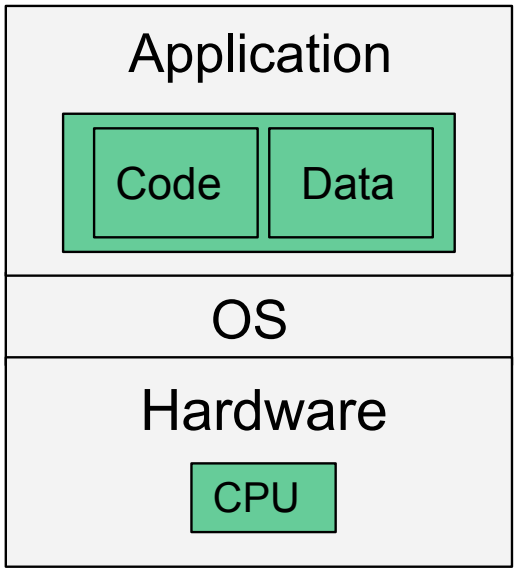
Trusted Execution Environment (TEE)

- Securely running sensitive application
  - **Isolated, always** encrypted enclaves
- Administrator
  - **Cannot** influence or monitor the application
- Authentication, Integrity, Confidentiality, Privacy



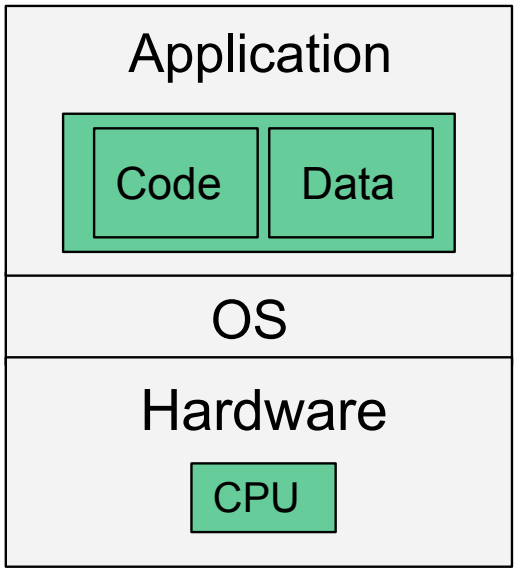
Trusted Execution Environment (TEE)

- Securely running sensitive application
  - **Isolated, always** encrypted enclaves
- Administrator
  - **Cannot** influence or monitor the application
- Authentication, Integrity, Confidentiality, Privacy
- Trust the CPU vendor

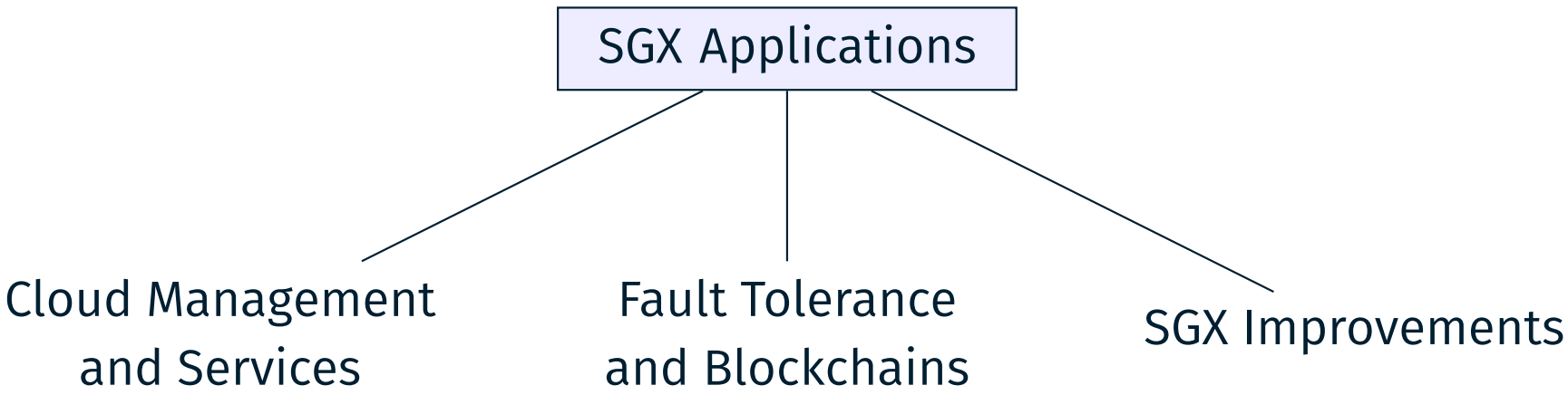


Trusted Execution Environment (TEE)

- Securely running sensitive application
  - **Isolated, always** encrypted enclaves
- Administrator
  - **Cannot** influence or monitor the application
- Authentication, Integrity, Confidentiality, Privacy
- Trust the CPU vendor
- Remote attestation is crucial
  $\rightarrow$ Trust for attestation service
- Sealing for persistent encryption
  $\rightarrow$ rollback attacks



Use cases

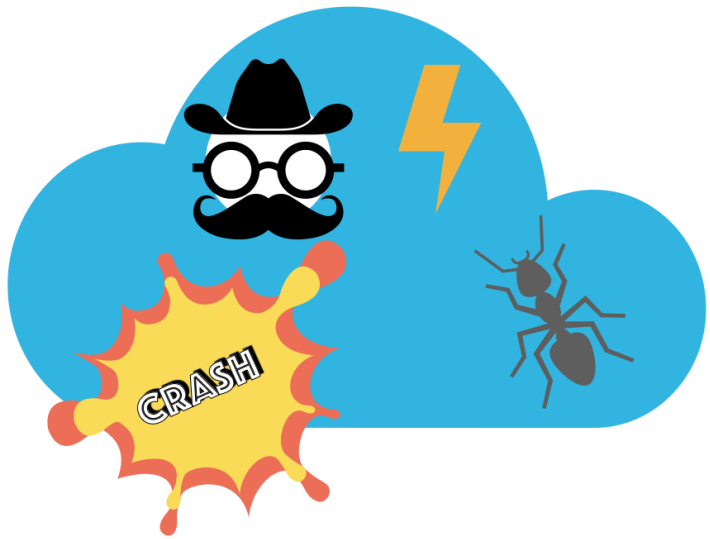

SGX Applications
Cloud Management and Services
Fault Tolerance and Blockchains
SGX Improvements

Byzantine Faults

- Provide a service despite arbitrary faults

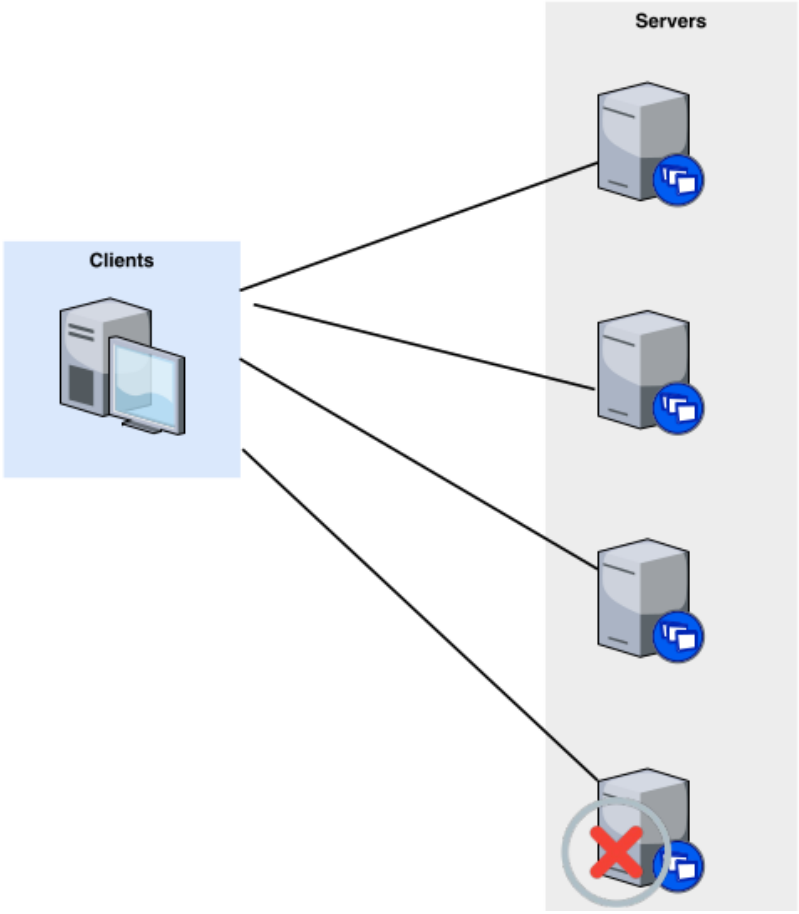
  $\rightarrow$ Byzantine Fault include:

  - Crashes, timing, and network failures
  - Misconfiguration
  - Software bugs
  - Attackers controlling part of the system
  - Participants sabotaging the system



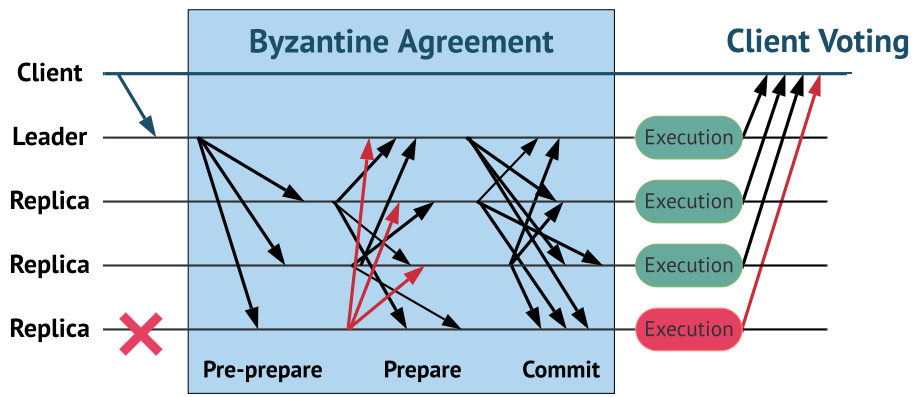
State Machine Replication

- Use multiple replicas
  - All contain a copy of the service
  - Keep replicas in sync
  - Keep functioning despite failures



Ordering

- Ordering algorithm that tolerates **Byzantine faults**
  - A leader proposes an order
  - Replicas need to confirm the order
- Faulty leader may propose wrong $\rightarrow$ suspect and change leader
- Hybrid protocols
  - BFT with TEEs
  - Prevent equivocation
  - Less communication rounds
- **Practical Byzantine Fault Tolerance (PBFT)**



Fault Model and properties

System Properties:

- Safety: all servers execute the same sequence of requests
- Liveness: all correct requests are eventually executed

Assumptions

- How many faults? And what type? (Crash tolerant, byzantine..)
- What timing assumptions:
  - E.g., upper bound on the execution of a request
  - Synchrony, asynchrony and partial synchrony
- What other assumptions you paper focuses on?

Organization and Research

Organisation

- Essay and presentation in **English**
  - Slides *and* voice track
- Certificate Requirements
  - Essay (6 pages, double column)
  - Presentation of own topic (20min + discussion)
  - Review
  - Active participation in discussions
- **Regular meeting: TBA**
  - You have to attend every meeting (Anwesenheitspflicht)
  - If you have an excuse, write an Email before

Time Management

- This seminar also covers
  - To work independently
  - Time management
  - Your curiosity about the topic

→ **You** are responsible for your deadlines:

  - Meeting with supervisor
  - Essay + Dry-Run
  - Presentation
  - Final Essay
  - Review for Topic x

Meeting with supervisor

- Carefully reading and analyzing your paper and technologies

- What to bring:
  - **Questions** regarding your papers and anything else!
  - Current status of your essay:
    - Minimal requirement: Outline
    - Better: Bullets or notes for every sections
    - Best: Draft text for some sections

- Goals of the meeting:
  - Help you to understand your topic better
  - Give hints for your essay and presentation

How to Read a Paper?

- **The first pass**
  - Carefully read the title, abstract, and introduction
  - Section and sub-section headings
  - Conclusion
- **The second pass**
  - Figures, diagrams
  - Special attention to **graphs and evaluation**!
- **The third pass**
  - This pass requires great attention to detail
  - Identify and challenge every assumption in every statement

Seminar Essay

Requirements Essay (1/2)

- 6 pages (ACM Proceedings template)
- LaTeX is required!
- Use the ACM SIGPLAN template
  - Download: `https://www.acm.org/publications/proceedings-template`
  - File to use: `sample-sigplan.tex`
- **Submit to:**
  `akss-betreuer@lists.informatik.uni-erlangen.de`

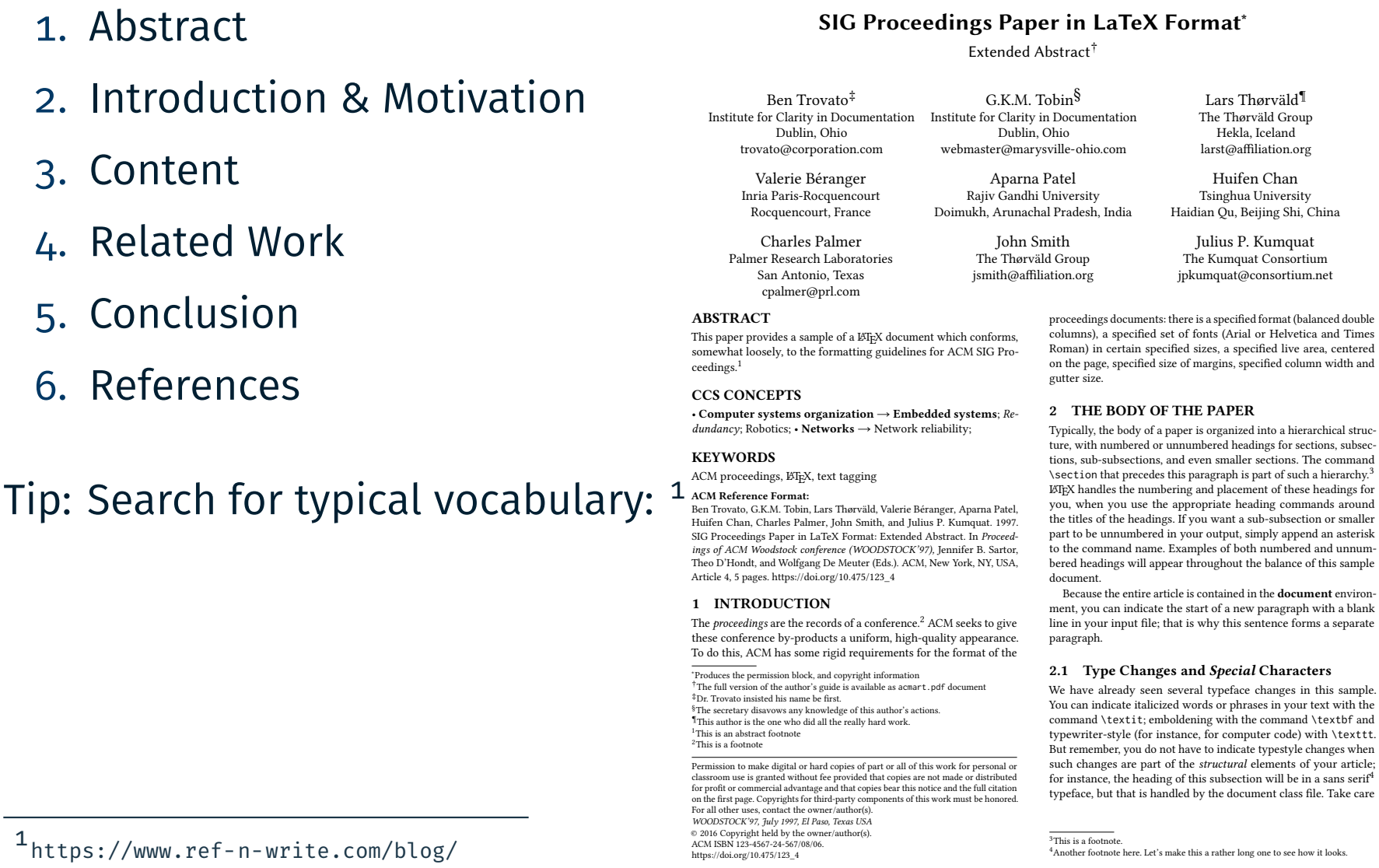
Your task:

- Write a **summary** of a paper
- Typical common mistakes:
  - Only including one paper as a reference
  - Repeat it everywhere in the essay

Requirements Essay (2/2)

1. Abstract
2. Introduction & Motivation
3. Content
4. Related Work
5. Conclusion
6. References

Tip: Search for typical vocabulary: [1]



SIG Proceedings Paper in LaTeX Format*

Extended Abstract†

Ben Trovato‡
Institute for Clarity in Documentation
Dublin, Ohio
trovato@corporation.com

G.K.M. Tobin§
Institute for Clarity in Documentation
Dublin, Ohio
webmaster@marysville-ohio.com

Lars Thørväld¶
The Thørväld Group
Hekla, Iceland
larst@affiliation.org

Valerie Béranger
Inria Paris-Rocquencourt
Rocquencourt, France

Aparna Patel
Rajiv Gandhi University
Doimukh, Arunachal Pradesh, India

Huifen Chan
Tsinghua University
Haidian Qu, Beijing Shi, China

Charles Palmer
Palmer Research Laboratories
San Antonio, Texas
cpalmer@prl.com

John Smith
The Thørväld Group
jsmith@affiliation.org

Julius P. Kumquat
The Kumquat Consortium
jpkumquat@consortium.net

ABSTRACT

This paper provides a sample of a LaTeX document which conforms, somewhat loosely, to the formatting guidelines for ACM SIG Proceedings.[1]

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

ACM proceedings, LaTeX, text tagging

ACM Reference Format:

Ben Trovato, G.K.M. Tobin, Lars Thørväld, Valerie Béranger, Aparna Patel, Huifen Chan, Charles Palmer, John Smith, and Julius P. Kumquat. 1997. SIG Proceedings Paper in LaTeX Format: Extended Abstract. In *Proceedings of ACM Woodstock conference (WOODSTOCK'97)*, Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 4, 5 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

The *proceedings* are the records of a conference.[2] ACM seeks to give these conference by-products a uniform, high-quality appearance. To do this, ACM has some rigid requirements for the format of the proceedings documents: there is a specified format (balanced double columns), a specified set of fonts (Arial or Helvetica and Times Roman) in certain specified sizes, a specified live area, centered on the page, specified size of margins, specified column width and gutter size.

2 THE BODY OF THE PAPER

Typically, the body of a paper is organized into a hierarchical structure, with numbered or unnumbered headings for sections, subsections, sub-subsections, and even smaller sections. The command `\section` that precedes this paragraph is part of such a hierarchy.[3] LaTeX handles the numbering and placement of these headings for you, when you use the appropriate heading commands around the titles of the headings. If you want a sub-subsection or smaller part to be unnumbered in your output, simply append an asterisk to the command name. Examples of both numbered and unnumbered headings will appear throughout the balance of this sample document.

Because the entire article is contained in the **document** environment, you can indicate the start of a new paragraph with a blank line in your input file; that is why this sentence forms a separate paragraph.

2.1 Type Changes and *Special* Characters

We have already seen several typeface changes in this sample. You can indicate italicized words or phrases in your text with the command `\textit`; emboldening with the command `\textbf` and typewriter-style (for instance, for computer code) with `\texttt`. But remember, you do not have to indicate typestyle changes when such changes are part of the *structural* elements of your article; for instance, the heading of this subsection will be in a sans serif[4] typeface, but that is handled by the document class file. Take care

*Produces the permission block, and copyright information
†The full version of the author's guide is available as `acmart.pdf` document
‡Dr. Trovato insisted his name be first.
§The secretary disavows any knowledge of this author's actions.
¶This author is the one who did all the really hard work.
[1]This is an abstract footnote
[2]This is a footnote
[3]This is a footnote.
[4]Another footnote here. Let's make this a rather long one to see how it looks.

WOODSTOCK'97, July 1997, El Paso, Texas USA

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

[1] https://www.ref-n-write.com/blog/research-paper-sample-writing-introduction-section-academic-phrasebank-vocabulary/

Feedback for Essays

- There will be feedback for your essays:
  - **1 Review** from seminar participants
  - Writing reviews for systems conferences (Timothy Roscoe, 2007) `https://people.inf.ethz.ch/troscoe/pubs/review-writing.pdf`
- Reviews are based on a *review template*
- Tips for a good grade
  - Detailed review
  - Questions related to the essay

- **Submit review to:** `akss-betreuer@lists.informatik.uni-erlangen.de`

Review Template

```
Please answer the following questions inside this file.
It is not needed to use LaTeX or other text processing tools.
In total there are 6 questions, but the last question will be only
    visible for your seminar supervisors.

=====
1) Give a short summary of the essay in at most seven (7) sentences.


=====
2) What did you like about the essay?


=====
3) What did you NOT like about the essay? Please give suggestions on how
    to improve the essay.
```

Debugging your essay: Citations

- Do not use citations as a noun
  - If you remove the citation, the sentence should still be grammatically correct and complete
  - Example: ”[A072] contains a definition of...” $\rightarrow$ Wrong
- Spacing
  - Use a non-breaking space “~” between a citation and the preceding word
  - Example: ” Fault-tolerant protocols~`\cite{castro1999practical}`”
- Multiple citations
  - Use `\cite{key1,key2}`
  - Do not use `\cite{key1}` `\cite{key2}`
- Avoid multiple entries of the same paper

Seminar Presentation

Requirements Presentation

- 20 mins talks $\rightarrow$ about 20 slides
- Be prepared for discussion
- i4-Beamertemplate (i4neo)

- Structure of presentation (recommendation)
  - Introduction, Motivation
  - Problem
  - Approach
  - Evaluation, Conclusion (one slide summary!)

- LaTeX is nice but other templates are also accepted
- **Submit to:**
  `akss-betreuer@lists.informatik.uni-erlangen.de`
  (after presentation)

General Structure

- Title slide, author, overview of the topic (First slide)
- Motivation (Which problem is solved? And why?) (3 slides)
- Outline of the talk (1 slide)
- Necessary background (1-3 slides)
- Problem and solution description (5 slides)
- Evaluation (0-4 slides)
- My opinion? (1-2 slides)
- Summary / Conclusion (1 slide)
- Backup (0-∞ slides)

Presentation: Tips

- Title, author, page numbers on each slide
- Details are important
  - Consistent slide numbers
  - Correct spelling
- Images are better than text!
  - But: don’t overload the viewers!
  - No Images without explanation
- Not more than 7 main bullets per slide
- Emphasis by or *text*

Topics

Hybrid Linear Protocols

- Hybrid linear communication protocol
- Two trusted services: checker and accumulator
- Increase resilience
- Reduced communication complexity and latency
- What challenges exist with these protocols
- How the two trusted services are used?

- **Topic 1:**
  - DAMYSUS: streamlined BFT consensus leveraging trusted components (Decouchant et al., Eurosys'22)

Confidential Replicated Services

- Framework to build permissioned confidential blockchains
- Records evidence to blame TEEs that deviate from the protocol
- Understand the guarantees and properties of TEEs
- Supports recovery, code updates

- **Topic 2:**
  - CCF: A Framework for Building Confidential Verifiable Replicated Services (Russinovich et al., published by microsoft)
  - Additional documentation: `https://microsoft.github.io/CCF/main/operations/recovery`

TEE Replication Model

- TEEs miss general means of ensuring the freshness of persistent state
- Restart-rollback (RR) fault model for replicating TEEs
  - Capture the possible fault behaviors of TEEs with external state
- How to replicate TEE applications with faulty behaviour?
- Adoption of the RR model with existing replication protocols

- **Topic 3:**
  - RR: A Fault Model for Efficient TEE Replication (Dinis et al., NDSS'23)

Provisioning

- Deployment off TEEs require active involvement of the application owner
  - Dynamic commissioning and decommissioning
- Seamless commissioning of TEE applications
- Byzantine Fault-Tolerant storage service storage
  - orchestrate enclave replication without the application owner
- What are the tradeoffs and the model?

- **Topic 4:**
  - ReplicaTEE: Enabling Seamless Replication of SGX Enclaves in the Cloud (Soriente et al., (EuroS&P))

Service Meshes

- Infrastructure for service-to-service communication
- Control plane configure the proxies
- Confidential service mesh
- Provisioning of certificates, configurations, and parameters
- Remote attestation of the entire cluster

- **Topic 5:**
  - MARBLERUN: The control plane for confidential computing (Published by Edgeless Systems)
  - Constellation: Confidential Kubernetes
  - Partially trusting the service mesh control plane (C.Adam et al, 2022, IBM)

- Untrusted parties run smart contracts over private data
- Contracts run off-ledger in secure enclaves using Intel SGX
  - Data confidentiality, integrity and data access policies

- **Topic 6:**
  - PDO: Private Data Object: an overview (Bowman et al, 2018, published by Intel)
  - Reflections on trusting distributed trust (Dauterman et al., Hotnets'22)

- Combining blockchains with TEEs
- Architecture separates consensus from execution
- Confidential preserving smart contracts with high scalability
- Identify and treat the pitfalls arising from TEEs and blockchains
  - Attack scenarios (e.g., TEE terminating)

- **Topic 7:**
  - Ekiden: A Platform for Confidentiality-Preserving, Trustworthy, and Performant Smart Contracts
    (Cheng et al, EuroS&P 2018)

Distributed Ledger and TEEs

- Using ledgers to enhance TEEs security
- TEEs limitations
  - Tamper with network communications
  - Preventing TEE from communicating with outside wold
  - Replay state
- New model with append-only ledger
- Parties keep a copy of the ledger to confirm a publication
- Three parties: client-side TEE, ledger loggers, host application

- **Topic 8:**
  - ELI: Giving State to the Stateless: Augmenting Trustworthy Computation with Ledgers
    (Kaptchuk et al, NDSS'19)

State Transfer in distributed setting

- Migration is hard to achieve
  - Operational inefficiencies and data loss
- Software-only migration functionality into existing TEE architectures
  - Preserving security guarantees
  - Interrupt TEE and migrate at any point
  - Stateful migration-related policies

- **Topic 9:**
  - CTR: Checkpoint, Transfer, and Restore for Secure Enclaves (Nakatsuka et al, 2022, ArXiv)

Secure Confidential Containers

- Lift-and-shift experience to run unmodified containers
- Run unmodified containers using AMD SEV-SNP processors
- Container-based technologies enable portable software deployment in the cloud
  - Container attestation and integrity
  - Container tampering is detected
  - Only the customer has access to its data
- **Topic 10:**
  - PARMA: Confidential Containers via Attested Execution Policies (A. Johnson et al, 2023, published by Azure Research ArXiv)

Rollback and Forking attacks

- Rollback or forking attacks
  - stale fata version or multiple versions
- Existing solutions suffer from performance overheads or weaker threat model
- Narrator proposes a blockchain based solution with TEEs
- achieves performance and state continuity
- **Topic 11:**
  - NARRATOR: Secure and Practical State Continuity for Trusted Execution in the Cloud (Niu et al, CCS'22)

TEE Raft

- Byzantine Fault Tolerant Protocol
  - Crash fault-tolerant protocols with TEEs
- Similar properties to BFT properties
- Attack vectors of SGX
  - Recovery and rollback

- **Topic 12:**
  - ENGRAFT: Enclave-guarded Raft on Byzantine Faulty Nodes (Niu et al, CCS'22)

- Secure in-memory distributed storage system
  - strong security, fault-tolerance, consistency (linearizability)
- Comparable with BFT systems
- Confidentiality with fewer replicas
- Secure and crash-consistent persistency
  - Distributed rollback protection.

- **Topic 13:**
  - AVOCADO: A Secure In-Memory Distributed Storage System (Bailleu et al., ATC'21)
- **Topic 14:**
  - TREATY: Secure Distributed Transactions (Giantsidi et al., DSN'22)

Privacy Attacks

- New concerns in access pattern leakage and software upgrade mechanisms
- Review of a cohort of four TEE-based smart contract platforms
- First replay and access pattern attacks on in-use TEE-based smart contract systems

- **Topic 15:**
  - SGXonerated: Finding (and Partially Fixing) Privacy Flaws inTEE-based Smart Contract Platforms Without Breaking the TEE

Better Hybrid Protocols

- Existing trust-bft protocols are not efficient
- Loss of Safety under Rollback and responsiveness problem
- FlexiBFT is a more efficient hybrid protocol
- Supports recovery

- **Topic 16:**
  - FlexiBFT: Dissecting BFT Consensus: In Trusted Components we Trust! (Gupta et al, Eurosys'23)

Better Hybrid Protocols

- Blockchains add overhead
- Payment networks enable off-chain transaction exchange
- Brings new attack vectors where parties steal funds
- TEEChain establish secure off-chain payment channels
- New variant of chain replication with threshold secret sharing

- **Topic 17:**
  - TEECHAIN: A Secure Payment Network with Asynchronous Blockchain Access (Lind et al, SOSP'19)

1. Damysus
2. CCF
3. RR
4. ReplicaTEE
5. Service Meshes
6. PDO
7. Ekiden
8. ELI
9. CTR
10. Parma
11. Narrator
12. Engraft
13. Avocado
14. Treaty
15. SGXonerated
16. FlexiBFT
17. TEEChain