How to Write a Good Paper

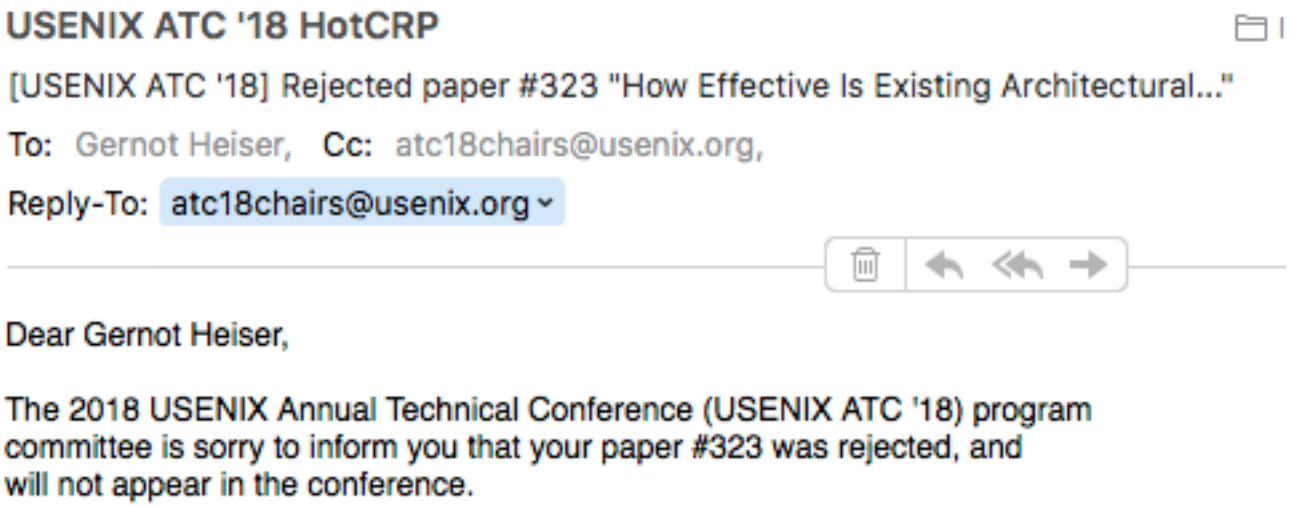
... or what I wish my supervisor had taught me

Gernot Heiser | gernot.heiser@data61.csiro.au | @GernotHeiser

TS Bootcamp May'18

www.data61.csiro.au

CSIRO


USENIX ATC '18 HotCRP
[USENIX ATC '18] Rejected paper #323 "How Effective Is Existing Architectural..."
To: Gernot Heiser, Cc: atc18chairs@usenix.org,
Reply-To: atc18chairs@usenix.org
Dear Gernot Heiser,
The 2018 USENIX Annual Technical Conference (USENIX ATC '18) program committee is sorry to inform you that your paper #323 was rejected, and will not appear in the conference.

Rejection Is Life

- My 2013 stats (my second-best year ever!):
  - 11 accepts:
    - 6 conferences: EuroSys, SIGMOD, SOSP, OOPSLA, 2*RTAS
    - 4 workshops: HotOS, APSys, PLOS, HotPower,
    - 1 journal: TOCS (plus TODS invite)
  - 8 rejects: 2*Usenix, PLDI, 2*RTSS, APSys, EMSOFT, RTAS
- My 2017 stats (a bad year):
  - 4 accepts:
    - 1 conference: EuroSys (paper rejected 5 times!)
    - 2 workshops: PLOS, APSys
    - 1 magazine (invited): IEEE Design & Test
  - 7 rejects:
    - Usenix Security, IEEE S&P × 2, RTAS, ASPLOS, SOSP × 2

Qualifications?

- Served on all top-tier conference TPCs in my field
  - at least one top-tier PC per year
- Presently
  - associate editor of IEEE Transactions on Computers
  - EB member of ACM Communications Research Highlights

Gernot Heiser

FOLLOW

Professor of Computer Science, UNSW Sydney, and Data61, CSIRO

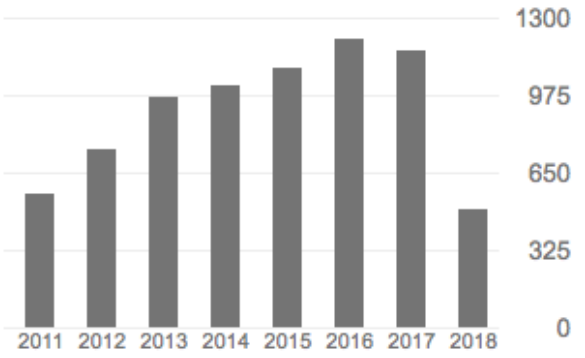
Verified email at unsw.edu.au - Homepage

Operating Systems Embedded Systems Security and Trustworthiness Energy Management Real-Time Systems

TITLE	CITED BY	YEAR
seL4: Formal verification of an OS kernel G Klein, K Elphinstone, G Heiser, J Andronick, D Cock, P Derrin, ... Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles ...	1517	2009
An Analysis of Power Consumption in a Smartphone. A Carroll, G Heiser	1395	2010

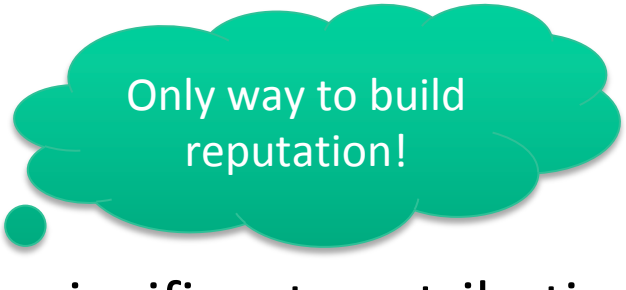
Cited by VIEW ALL

	All	Since 2013
Citations	10062	5972
h-index	48	34
i10-index	115	70



Ways to Avoid Rejection

- Safe way: Aim low:
  - 2nd/3rd-tier venues are easy
  - *guaranteed impact-free*



- Hard way: Write an excellent paper with a significant contribution

THIS ADVICE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Journals vs Conferences

- I'm in Computer Science, sub-discipline Operating Systems
  - 90% of top OS publications are in conferences
  - other sub-disciplines are different: CS theory is about 50% journals
- CS top conference reviewing much tougher than journals
- Non-CS conferences are more lightweight, journals dominate

	Computer Science Especially systems	Other
Complete, significant work	Top-tier conference	Transactions
Less significant, recycled rejects	Secondary conference, journal	Other journal
Early work, position papers	Workshop	Conference

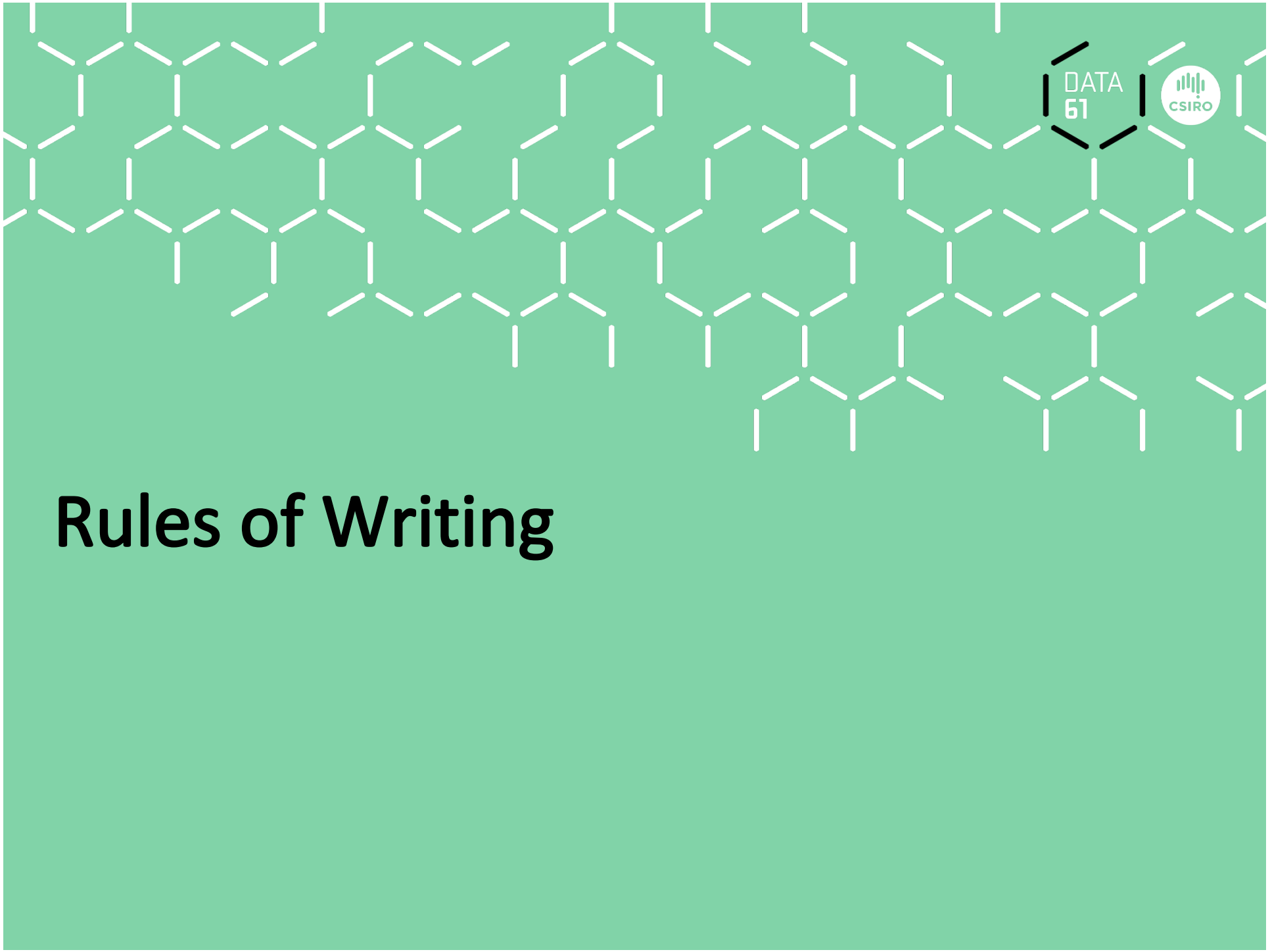
What is “Systems”?

(Overly?) simplified view of Computer science: theory + systems

- Theorists build theories, models
  - often get away with theories not good for anything
- Systems folks build stuff
  - don’t get away with work not good for anything!

Examples of “systems” work:

- operating systems
- network systems / distributed systems
- database systems
- programming systems (PL implementation)
- machine-learning systems
- …



Rules of Writing

Rule 1: Reviewers are Pot Luck

- … even at top conferences
  - even good papers get rejected, sometimes for the wrong reasons
  - … but most times rejection is your fault!

- Reviewers' top reasons for rejection
  - I'm not convinced you're solving a *real problem*
  - I'm not convinced you're *solving* the problem
  - *I don't understand* – your paper is too badly written
  - Insufficient contribution for {SOSP, OSDI, EuroSys…}

- Papers without a PC "champion" have a hard stand
  - Make sure there's something which at least one reviewer will think cool
  - Purely incremental work will have a hard stand at top venues

Rule 2: A Paper Has a Story

1. The paper has a (one!) main message
   - Understand clearly what the message is
   - Make sure that the reader gets it
   - Make sure it's an interesting one

2. A paper has a narrative
   - It starts from zero and then works on transmitting the message
   - *Everything* you write must support the message
   - *Maintain reader state!*
     - be conscious of what the reader knows/remembers

Rule 3: Limited Space: The Two “C”s

- Be *clear* (at all levels)
  - every sentence, paragraph, section has a clear purpose
  - the purpose is clearly communicated
  - the overall message is consistent
- Be *concise* (brief but complete)
  - don’t waffle!!! (Use “Jay’s rule of thumb”)
  - be precise
  - make sure it’s readable, lucid, enjoyable

But:

- maintain reader state:
  - define before use
  - be aware of what the reader has learned
  - recall/remind if necessary
- Make sure it’s self-contained

Rule 4: Presentation Matters

Top conferences tend to accept two kinds of papers

1. excellent work that is presented well
2. average work that is presented well

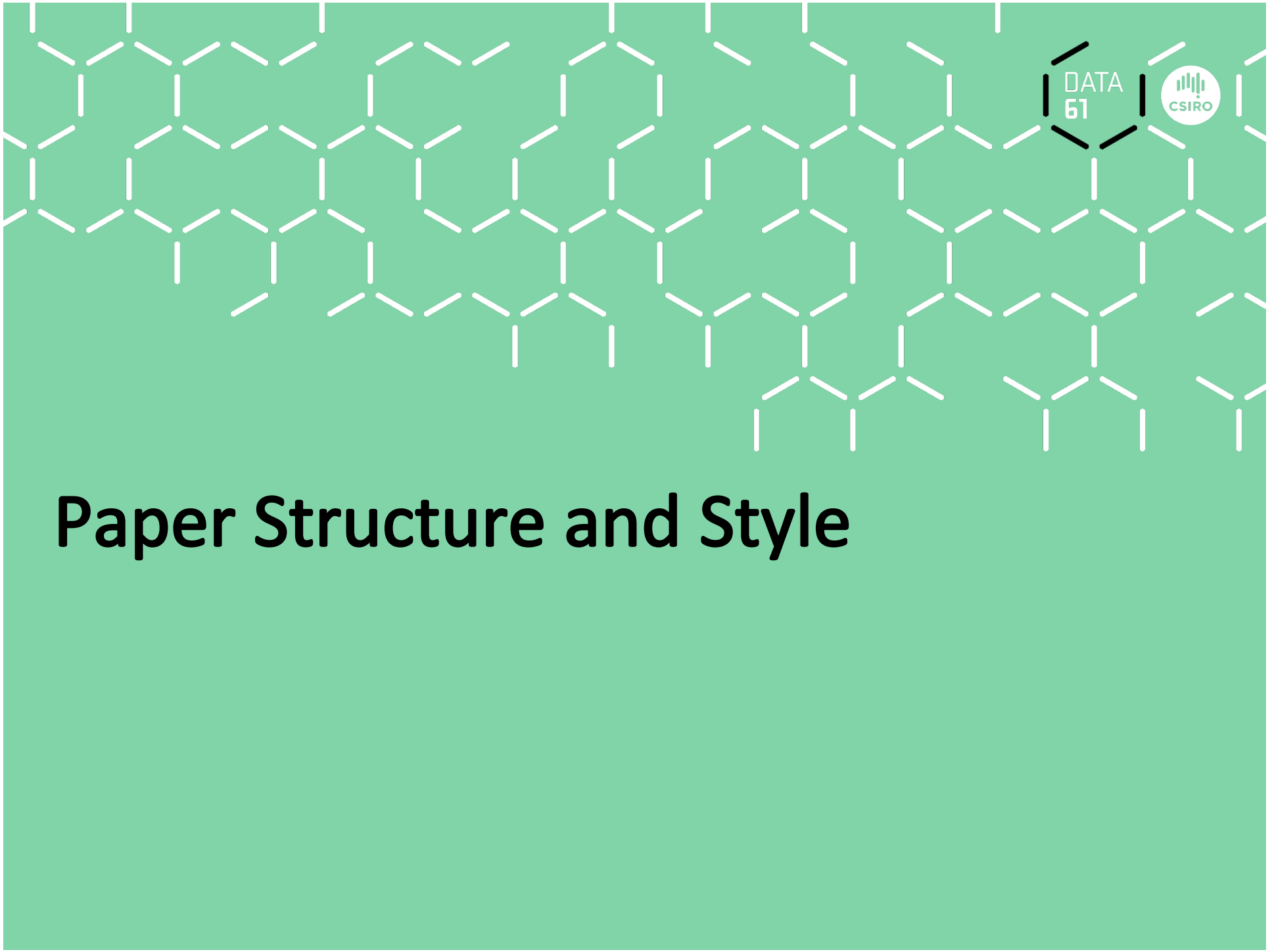
The best work is useless if you can't convince the reviewers

- reviewers are busy, may have to review 30 papers
- they'll look for reasons to reject – don't give them any!

Rule 4: Presentation Matters – Paper Engineering

Important bits:

- Introduction: sell the idea, the significance and the approach
- build tension, make reader interested
- convincing argumentation
- top-down, not bottom-up
- maintain reader state
- convincing evaluation
  - thorough and honest
- *state assumption/limitations honestly*



Paper Structure and Style

Introduction:
Most Important Part of the Paper!

The Overture:

- Explain the problem you're solving, why it's a problem
- Outline your approach
- Indicate results/outcomes
- State contributions
- "Paper roadmap" is a waste of space

General hints for intro:

- Capture the reader's interest: sell your idea
- Be concise: Stay within about one page!
- Make sure the paper delivers what you promise
  - Reviewers kill for "bait and switch"

Other Parts

- Background: set the scene in more detail
  - cite related work as needed, don’t discuss more than necessary
  - Examples!!!!
- Describe problem in detail
- Explain solution in detail
  - be honest and forthcoming with limitations and assumptions
- Evaluation: for systems work often largest part
- Related work
- Conclusions
- Abstract
  - used to steer to the right reviewers
  - What, Why, Achievement, Implication, one sentence each
  - IMPORTANT: Redo for camera-ready!

Evaluation

- Show that your solution actually works
  - *Progressive*: significant improvements in important situations
  - *Conservative*: no (significant) degradation elsewhere

  Need both!

Benchmarking Crimes (Selection)

1. Selective Benchmarking – cherry picking
2. Only micro-benchmarks
3. Throughput degradation = overhead
4. Creative overhead accounting
5. Improper baseline, only relative figures, compare against self
6. No indication of significance

Full list: http://gernot-heiser.org/benchmarking-crimes.html

Style and Form

- Write in engaging style, lead reader though the paper
  - avoid bottom-up structure, *present ideas top-down*
  - follow style rules
  - *Use active voice!!!!* … and present tense
  - Avoid buzzwords (“novel”, “mobile social post-quantum fog computing”)
- Be mindful of reader’s brain state (which is lossy)
  - *maintain reader state*
  - don’t assume every reviewer is expert in your narrow area
  - but don’t think you can hide stuff from reviewers!
- Follow formatting rules
  - don’t play with margin, baseline skip etc
  - don’t use microscopic fonts, >40y olds have problems with <8pt font
- Spell-check, proof-read, proof-read
  - get native speaker to proof-read if you aren’t
  - get outsider to read it – great way to spot holes before it’s too late!

Mechanics

- Use revision control
  - especially (but not only) when it's a joint paper
- Don't use MS Word
  - doesn't integrate well with revision control
  - requires coarse-grain locking
  - references are painful, formulae even more so
  - MSR people use LaTeX, so should you!
- Use BibTeX
  - … but use it correctly (eg capitalisation in titles)
- Use scriptable tools (eg GNUplot) for graphing results
  - Results change frequently and at the last minute
  - Being able to run from command line is essential

Summary

- Clear statement of problem
- Why would I care?
- Convincing solution, compelling argument
- Thorough evaluation, no BM crimes
- Lucid writing, maintaining reader state

Further Reading

Writing systems papers:

- Levin & Redell: An evaluation of the 9th SOSP submissions, or How (and how not) to write a good systems paper
- Simon Peyton Jones (MSRC): How to write a great research paper
  - http://research.microsoft.com/en-us/um/people/simonpj/papers/giving-a-talk/giving-a-talk-slides.pdf
- My paper/thesis writing guide
  - http://gernot-heiser.org/style-guide.html

General writing/style etc (recommended by systems folks):

- Zobel: Writing for computer science, Springer
- Strunk & White: The elements of style, Allyn & Bacon
- Dupré: Bugs in writing: A guide to debugging your prose, Addison-Wesley

DATA 61

Thank you

www.csiro.au