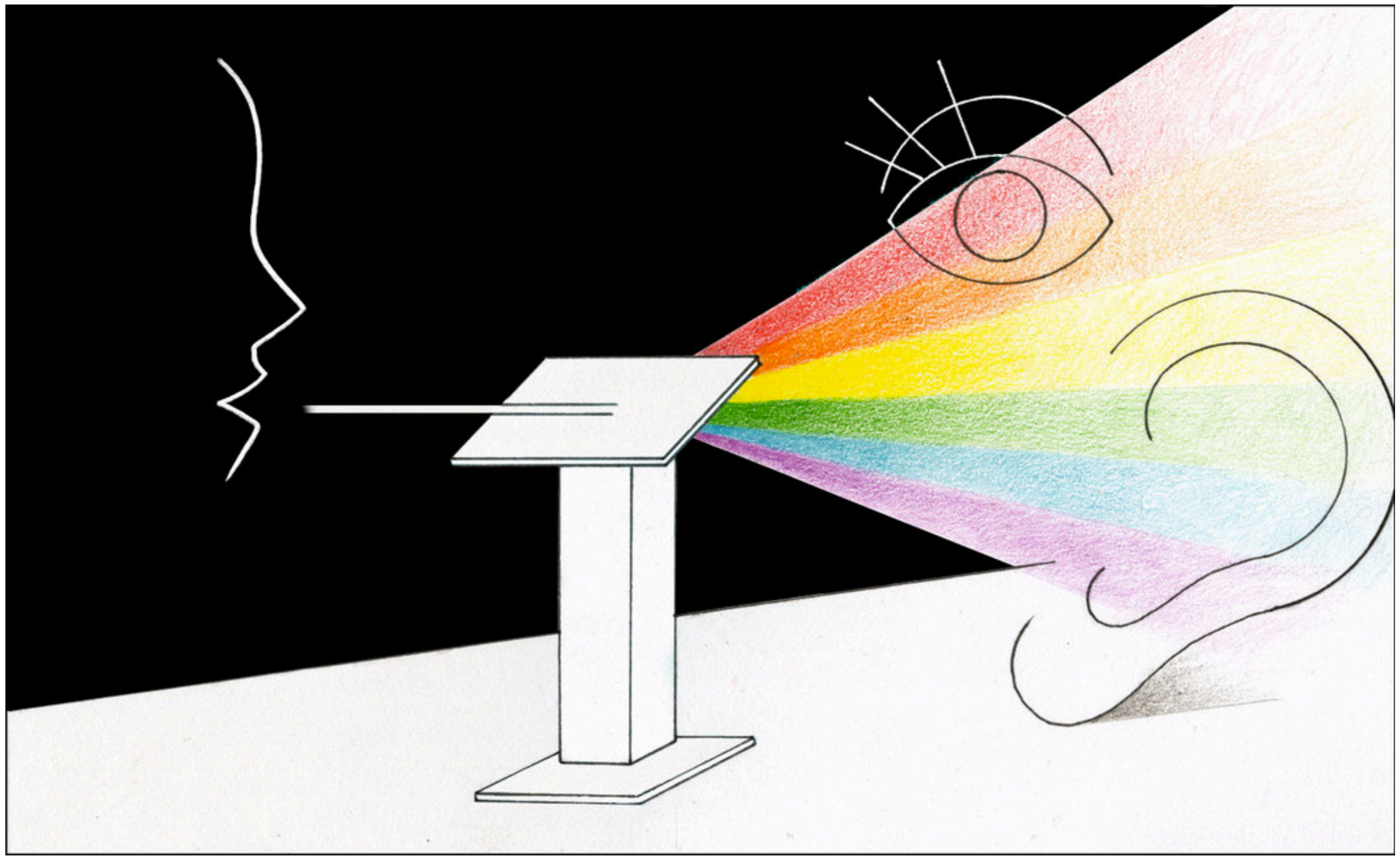
How to Give Talks That People Can Follow

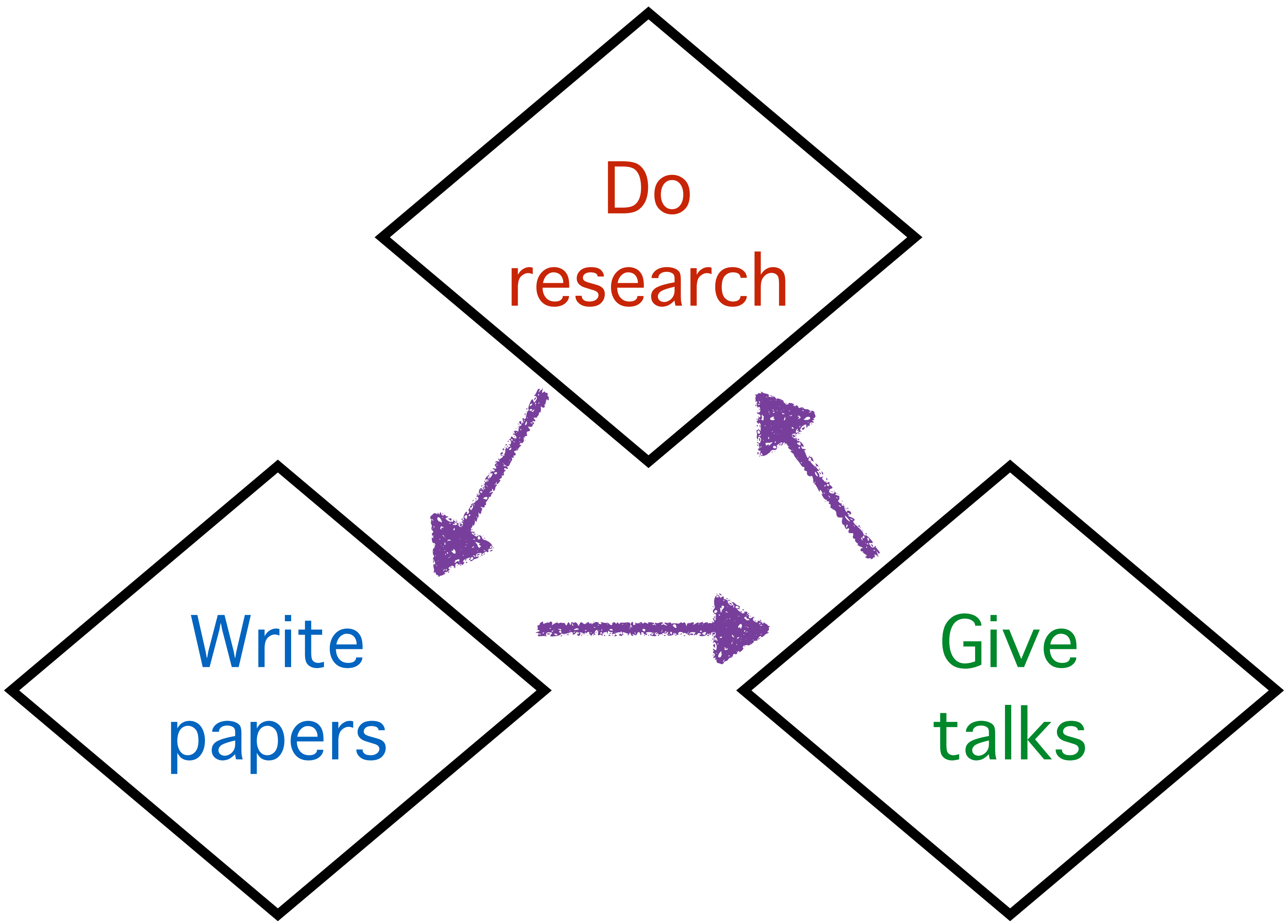


Derek Dreyer
MPI for Software Systems

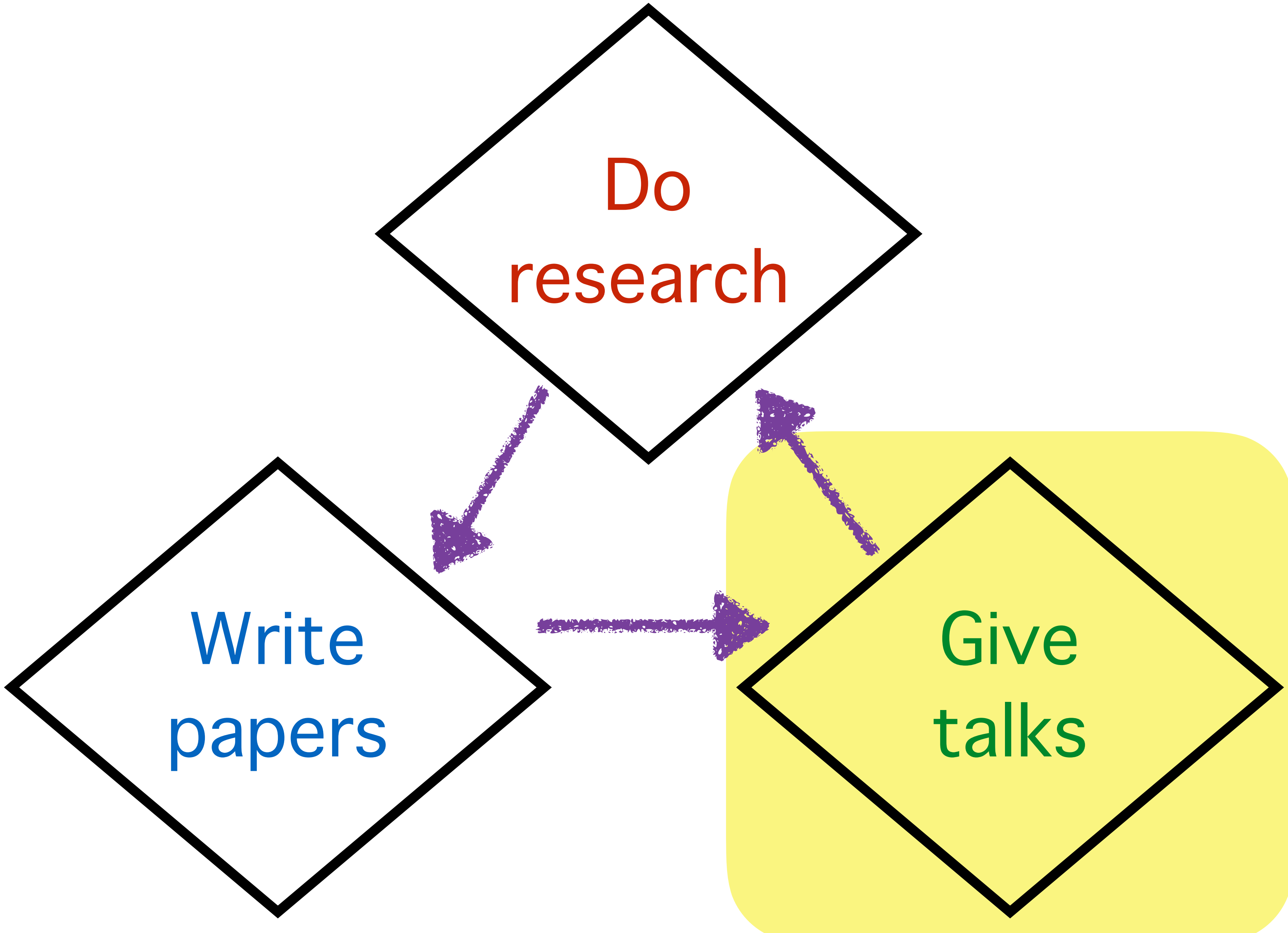
PLMW@POPL 2018
Los Angeles

My job as a researcher

My job as a researcher



My job as a researcher



Entertain your audience!

- **Simon Peyton Jones.** *How to give a great research talk*. (MSR Summer School, 2016)
  - "Your mission is to **wake them up**!"
  - "Your most potent weapon, by far, is **your enthusiasm**!"

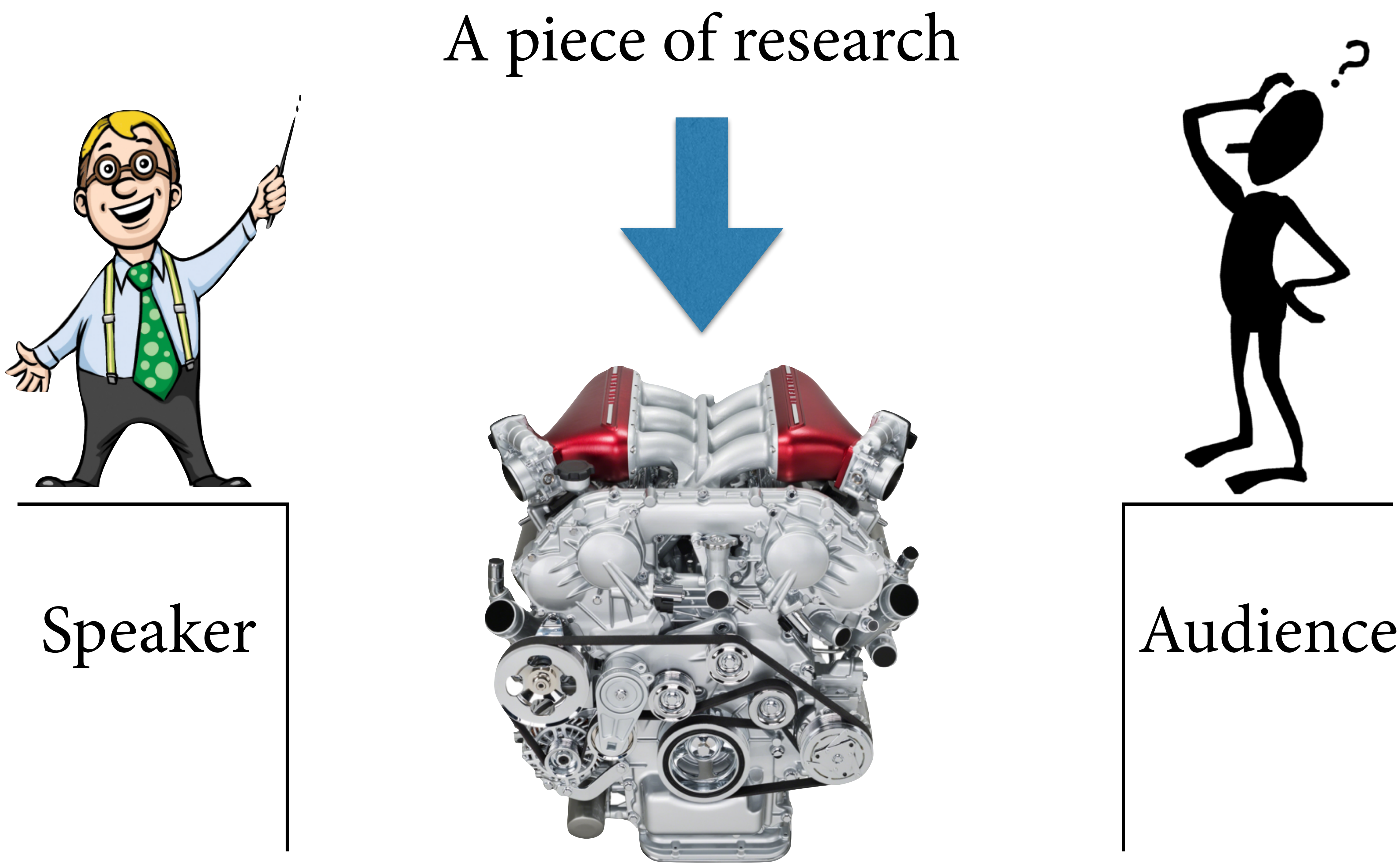
- **John Hughes.** *Unaccustomed as I am to public speaking*. (PLMW, 2016)
  - "**Put on a show**!"

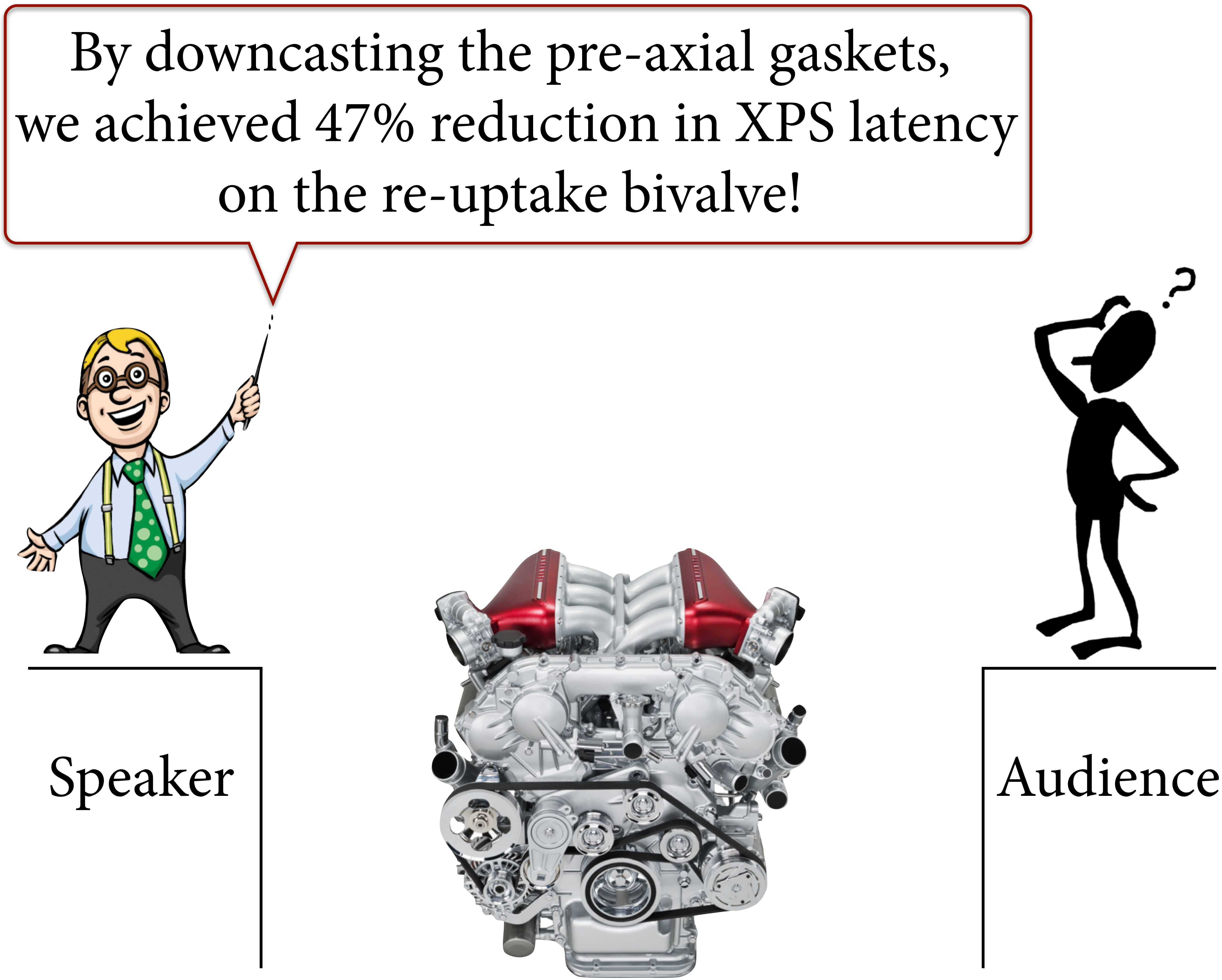
Entertain your audience!

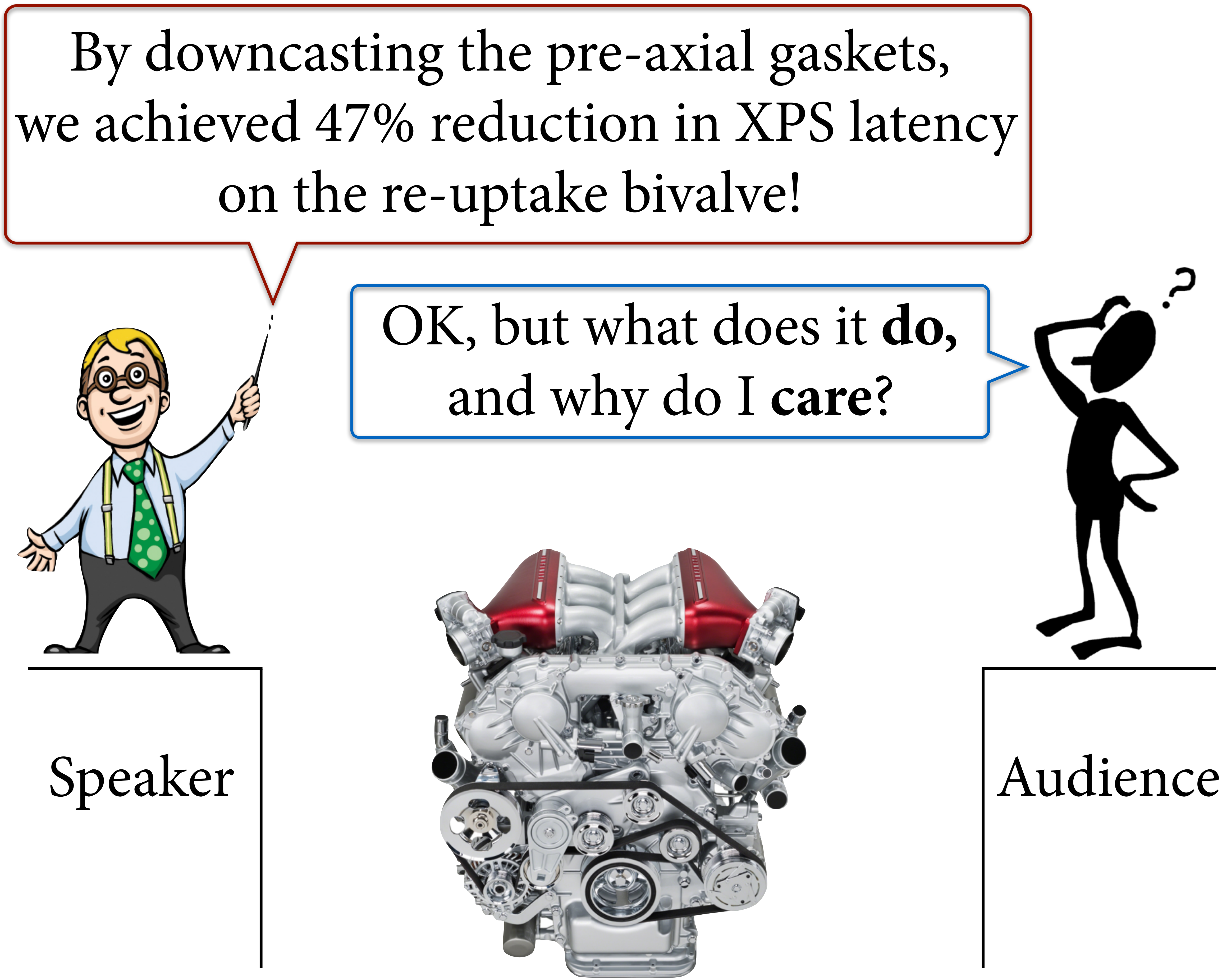
- **Simon Peyton Jones.** *How to give a great research talk*. (MSR Summer School, 2016)

Good advice, but I don't know how to teach people to be entertaining…

- **John Hughes.** *Unaccustomed as I am to public speaking*. (PLMW, 2016)
  - "**Put on a show**!"


A piece of research
Speaker
Audience


By downcasting the pre-axial gaskets,
we achieved 47% reduction in XPS latency
on the re-uptake bivalve!
Speaker
Audience


By downcasting the pre-axial gaskets, we achieved 47% reduction in XPS latency on the re-uptake bivalve!
OK, but what does it **do,** and why do I **care**?
Speaker
Audience

How is a conference talk different from a paper?

Conference talks

On the plus side:

- ✓ Lots of eyeballs on you and your work!

On the minus side:

Conference talks

On the plus side:

- ✓ Lots of eyeballs on you and your work!

On the minus side:

- ✗ You can't say much.
- ✗ The audience may or may not care.
- ✗ Even those who care will easily get lost.

Conference talks

On the plus side:

- ✓ Lots of eyeballs on you and your work!

On the minus side:

- ✗ You can’t say much.
- ✗ The audience may or may not care.
- ✗ Even those who care will easily get lost.

A paper structure that works

- **Abstract**
- **Intro**
- **Key ideas**
- **Technical meat**
- **Related work**

A paper structure that works

- **Abstract**
- **Intro**
- **Key ideas**
- **Technical meat**
- **Related work**

talk

A ~~paper~~ structure that works

- ~~**Abstract**~~
- **Intro**
- **Key ideas**
- ~~**Technical meat**~~
- ~~**Related work**~~

talk

A ~~paper~~ structure that works

- ~~Abstract~~
- Intro
- Key ideas
- ~~Technical meat~~
- ~~Related work~~

A ~~paper~~ talk structure that works

- **Intro** (8 minutes)
- **Key ideas** (11 minutes)

A ~~paper~~ talk structure that works

- **Intro** (8 minutes)
- **Key ideas** (11 minutes)
- **What else is in the paper** (1 minute)

Key ideas

- Use **concrete illustrative examples** and high-level intuition.
- Do **not** show the general solution!
  (People can go read your paper for that.)

Conference talks

On the plus side:

- ✓ Lots of eyeballs on you and your work!

On the minus side:

- ✗ You can't say much.
- ✗ The audience may or may not care.
- ✗ Even those who care will easily get lost.

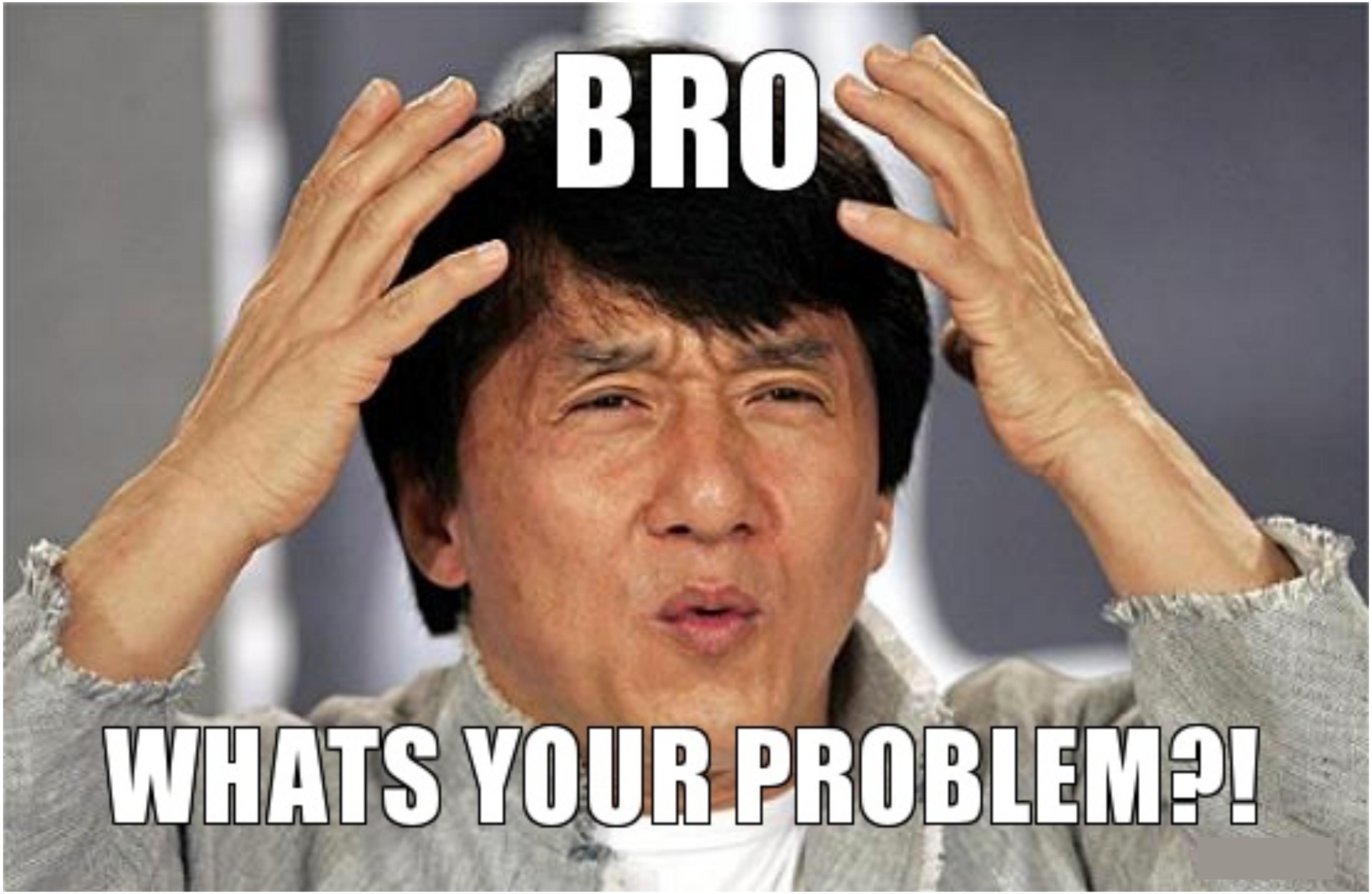
Conference talks

On the plus side:

- ✓ Lots of eyeballs on you and your work!

On the minus side:

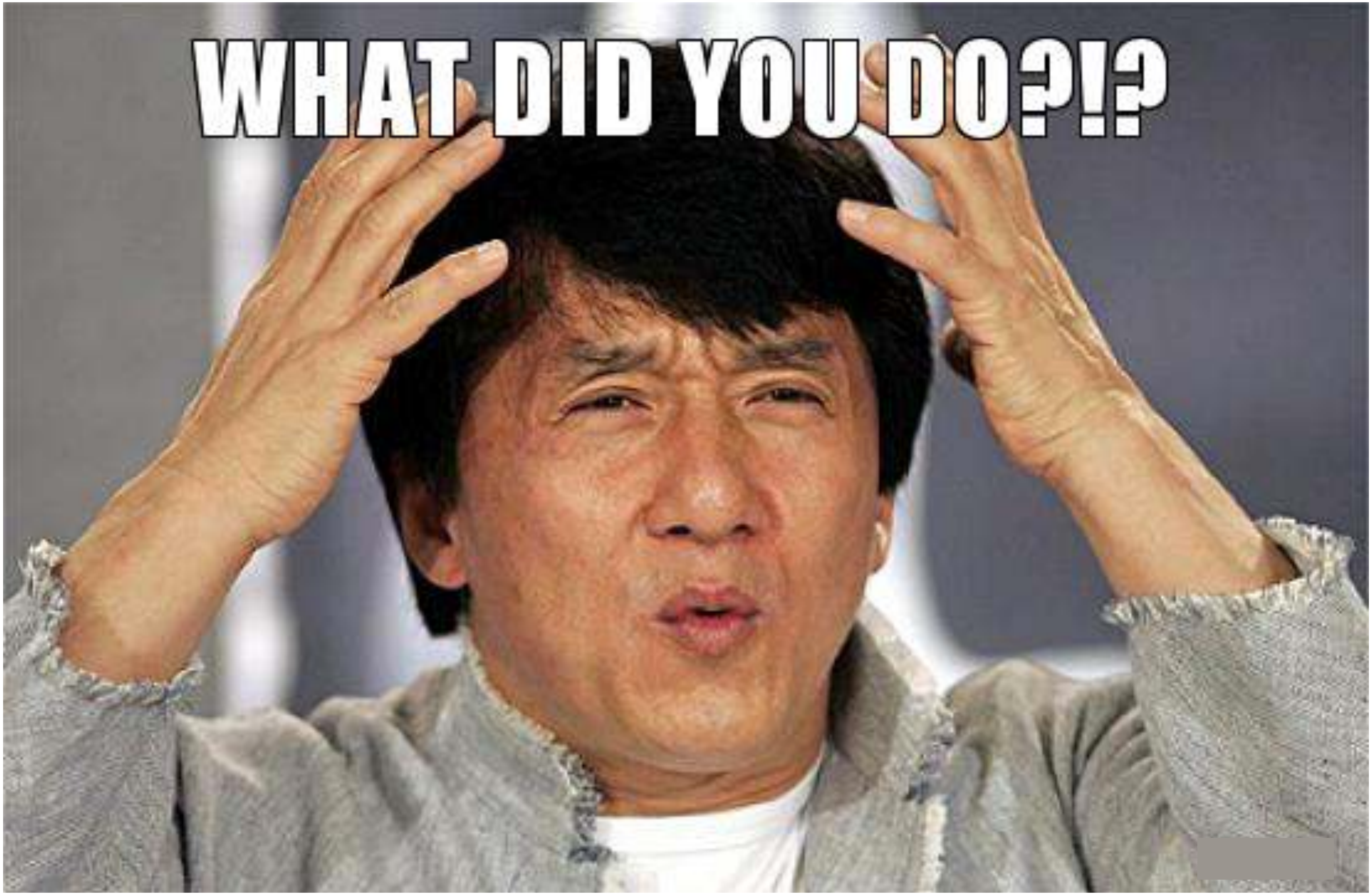
- ✗ You can't say much.
- ✗ The audience may or may not care.
- ✗ Even those who care will easily get lost.


BRO
WHATS YOUR PROBLEM?!

Stage the motivation

- **First, get to a problem.**
  - Explain a **general** version of your problem (but not too general) **in the first 2 minutes**.

- **Then, get to the problem.**
  - Motivate and **explicitly state** your **specific** problem in the next 4 minutes.
  - Limit discussion of prior work only to what is needed to explain your problem.


WHAT DID YOU DO?!?

Tell them what you did!

- **Proudly state your contributions.**
  - After the motivation, the audience eagerly wants to hear what you did. Tell them!

- **Follow immediately with a crisp statement of your key idea(s).**
  - It will give audience a take-home message, and give focus to the rest of your talk.

Conference talks

On the plus side:

- ✓ Lots of eyeballs on you and your work!

On the minus side:

- ✗ You can't say much.
- ✗ The audience may or may not care.
- ✗ Even those who care will easily get lost.

Conference talks

On the plus side:

- ✓ Lots of eyeballs on you and your work!

On the minus side:

- ✗ You can't say much.
- ✗ The audience may or may not care.
- ✗ Even those who care will easily get lost.

Flow & coherence

Create **flow** with **old to new**

Create **coherence** with **one slide, one point**

Does this text flow?

Does this text flow?

Security proofs of cryptographic protocols are crucial for the security of everyday electronic communication. However, these proofs tend to be complex and difficult to get right. The game-playing technique, originally proposed by Jones et al., follows a code-based approach where the security properties are formulated in terms of probabilistic programs, called games. This is a general design principle for cryptographic proofs to ease their management.

Does this text flow?

Security proofs of cryptographic protocols are crucial for the security of everyday electronic communication. However, these proofs tend to be complex and difficult to get right. The game-playing technique, originally proposed by Jones et al., follows a code-based approach where the security properties are formulated in terms of probabilistic programs, called games. This is a general design principle for cryptographic proofs to ease their management.

Does this text flow?

Security proofs of cryptographic protocols are crucial for the security of everyday electronic communication. However, these proofs tend to be complex and difficult to get right. The game-playing technique, originally proposed by Jones et al., follows a code-based approach where the security properties are formulated in terms of probabilistic programs, called games. This is a general design principle for cryptographic proofs to ease their management.

Does this text flow?

Security proofs of cryptographic protocols are crucial for the security of everyday electronic communication. However, these proofs tend to be complex and difficult to get right. The game-playing technique, originally proposed by Jones et al., follows a code-based approach where the security properties are formulated in terms of probabilistic programs, called games. This is a general design principle for cryptographic proofs to ease their management.

Does this text flow?

Security proofs of cryptographic protocols are crucial for the security of everyday electronic communication. However, these proofs tend to be complex and difficult to get right. The game-playing technique, originally proposed by Jones et al., follows a code-based approach where the security properties are formulated in terms of probabilistic programs, called games. This is a general design principle for cryptographic proofs to ease their management.

Does this text flow?

Security proofs of cryptographic protocols are crucial for the security of everyday electronic communication. However, these proofs tend to be complex and difficult to get right. The game-playing technique, originally proposed by Jones et al., follows a code-based approach where the security properties are formulated in terms of probabilistic programs, called games. This is a general design principle for cryptographic proofs to ease their management.

Does this text flow?

Security proofs of cryptographic protocols are crucial for the security of everyday electronic communication. However, these proofs tend to be complex and difficult to get right. The game-playing technique, originally proposed by Jones et al., follows a code-based approach where the security properties are formulated in terms of probabilistic programs, called games. This is a general design principle for cryptographic proofs to ease their management.

Does this text flow?

Security proofs of cryptographic protocols are crucial for the security of everyday electronic communication. However, these proofs tend to be complex and difficult to get right. The game-playing technique, originally proposed by Jones et al., follows a code-based approach

What does this game-playing technique have to do with what came before?

Old to new

- Begin sentences with old info
  - Creates link to earlier text
- End sentences with new info
  - Creates link to the text that follows
  - Also places new info in position of **emphasis**

Applying old-to-new

New information

Security proofs of cryptographic protocols are crucial for the security of everyday electronic communication. However, these proofs tend to be complex and difficult to get right. The game-playing technique, originally proposed by Jones et al., follows a code-based approach where the security properties are formulated in terms of probabilistic programs, called games. This is a general design principle for cryptographic proofs to ease their management.

Applying old-to-new

Security proofs of cryptographic protocols are crucial for the security of everyday electronic communication. However, these proofs tend to be complex and difficult to get right. To make it easier to manage such proofs, Jones et al. have proposed a new design principle, called the game-playing technique. This technique follows a code-based approach where the security properties are formulated in terms of probabilistic programs, called games.

Old-to-new satisfied

Security proofs of cryptographic protocols are crucial for the security of everyday electronic communication. However, **these proofs** tend to be complex and difficult to get right. To make it easier to manage **such proofs**, Jones et al. have proposed a new design principle, called the **game-playing technique**. **This technique** follows a code-based approach where the security properties are formulated in terms of probabilistic programs, called games.

Flow in talks

- **Within** a slide:
  - Script should follow "old to new"
- **Between** slides:
  - Don't just flip to next slide and say, "So…"
  - Plan something to say **during** the transition

Flow & coherence

Create **flow** with **old to new**

Create **coherence** with **one slide, one point**

Optimization & Concurrency

- Compiler performs several optimizations to generate optimized code.
    - >100 optimizations in GCC, LLVM.

Correct optimizations for sequential programs may be incorrect for shared memory concurrency.

State-of-the-Art:

- Compilers are over-conservative;
  * optimization opportunities are lost.

or

- Buggy optimization
  * *"Premature optimization is the root of all evil"* ~ Donald Knuth

Talklets

- **Break long stretches of talk into talklets.**
  - More digestible units of story (2-4 min.)
  - But just having talklets is not enough…
- **Use transitions between talklets to remind the audience of the big picture.**
  - Summarize the point of the last talklet and how it connects to the next one.

A few words about

SLIDE DESIGN

No sense of style?

Don't worry

The most important aspects of slide design have **nothing** to do with style

Access control is inadequate, scenario 2: Facebook timeline

- Facebook introduced timeline in 2011 end
  - Chronologically order all the information on your profile
  - Make them easily searchable for other users
- Easier to search Potentially embarrassing older content
- Users were afraid of privacy violation

Access control was not changed !

5

Access control is inadequate, scenario 3: Spokeo

- Service aggregating information about individuals
  - Each individual information is public content
  - E.g., your Facebook profile, address
- One can infer new non public information
  - Estimating wealth using address and public property records
- Users complain of privacy violation

Access control was not changed !

6

Access control is inadequate: Summary

- User reaction suggests each of the cases violate privacy
- However in none of the cases access control is violated
- We propose a new model to reason about privacy

7

Exposure : Definition

- We define *Prominence* of information I at time t or $P_I(t)$

$$P_I(t)=\{U \mid U \text{ is aware of I at time t}\}$$

- Then E_I ,exposure of I is:

$$E_I = \lim_{t\to\infty} P_I(t)$$

8

Modeling user privacy using exposure

- For each content users have an expected exposure
  - How many other users are likely to access the content
- We can model privacy violation for an information as
  - Large deviation of actual exposure from expected exposure

9

Revisiting scenario 1: Facebook newsfeed

- Before newsfeed was introduced
  - Expected exposure: Friends who will visit user's profile
  - Actual exposure was same as expected exposure
- After newsfeed was introduced
  - Actual exposure: All friends to whom the information is pushed
  - Actual exposure is much higher than the expected exposure

10

Revisiting scenario 2: Facebook timeline

- Before timeline was introduced
  - Expected exposure for older data: Friends who will scroll to find a old content
  - Actual exposure for older data was same as expected exposure
- After timeline was introduced
  - Actual exposure for older data: All friends who visit the profile
  - Actual exposure is much higher than the expected exposure

11

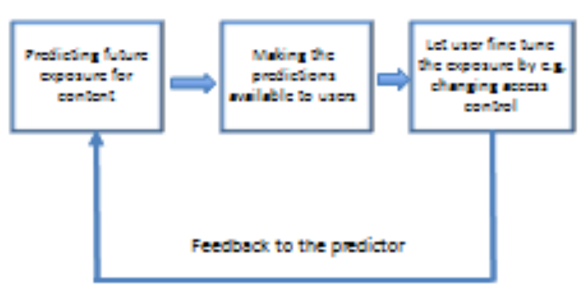
Revisiting scenario 3: Spokeo

- Before spokeo aggregated data
  - Expected exposure for new inferred data: Users who dig up each individual pieces of content form different sources
  - Actual exposure for older data was same as expected exposure
- After spokeo aggregated data
  - Actual exposure for new inferred data: All users who visit public spokeo website
  - Actual exposure is much higher than the expected exposure

12

Major Deviation from expected exposure can capture the privacy violations not covered by access control

Proposed model: managing privacy via exposure



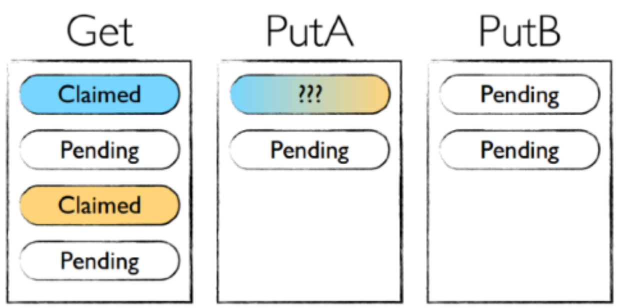
Key challenge: Predicting future exposure

- Huge existing work for predicting growth in content popularity
  - Future YouTube views, Facebook likes, Retweets
  - Use machine learning, regression techniques
  - We can leverage advances in those fields to predict exposure
- OSN operators are best positioned to do the predictions
  - Empirical data on how information disseminates in their sites
  - Facebook or Youtube already provide number of likes or views

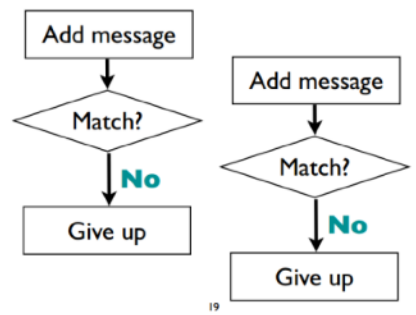
Limitations of our model

- Privacy violation by inference using available data
  - It is extremely hard to enumerate all possible inference
- Privacy violation using cross site prediction
  - Prediction across multiple systems
  - E.g., posting a picture taken from Facebook in tweeter

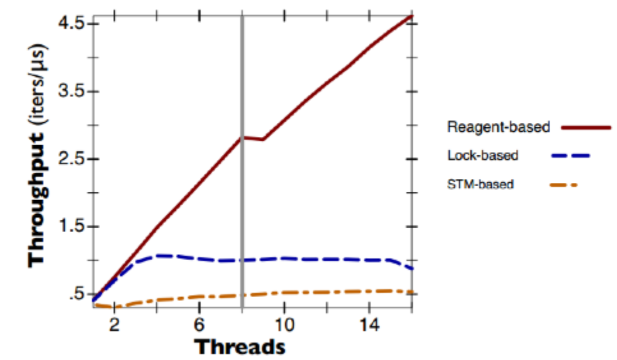
```
When(Get).And(PutA).And(PutB).Do(...)
```



The Protocol



Stack transfer



Our implementation (in C#)

Key idea:
Messages are resources

Store in lock-free bags
➡ parallelized matching
➡ decreased communication

Is this just STM?

Isolation	**Interaction**
Shared state	Message passing

Using lock-free bags, based on earlier work with Russo [OOPSLA'11]

The Problem:

Concurrency libraries are indispensable, but hard to build and extend

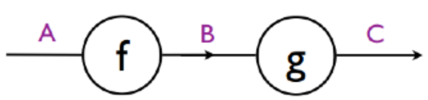
Joins: a crash course

```
class Lock {
  public Synchronous.Channel  Acquire = new ...
  public Asynchronous.Channel Release = new ...
  public Lock() {
    When(Acquire).And(Release).Do(() => {});

    // initially available
    Release();
  }
}
```

Join pattern

Lambda: the ultimate abstraction



This work

Use ***join patterns*** for synchronization: [Fournet & Gonthier]

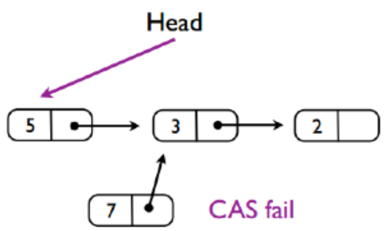
Expressive
Write synchronization primitives declaratively and concisely

Scalable
Competitive with industrial libraries; can recover existing algorithms

`java.util.concurrent`

Synchronization
- Reentrant locks
- Semaphores
- R/W locks
- Reentrant R/W locks
- Condition variables
- Countdown latches
- Cyclic barriers
- Phasers
- Exchangers

Data structures
- Queues
  - Nonblocking
  - Blocking (array & list)
  - Synchronous
  - Priority, nonblocking
  - Priority, blocking
- Deques
- Sets
- Maps (hash & skiplist)



State of the art?

Leave it to the experts:

Research Literature ⇔ Industrial-strength Libraries

java.util.concurrent
.NET 4.0
Intel TBB

Key takeaways

- **Avoid PowerPoint-itis**
  - Don't put lots of text on slides just so they are readable independently of the talk

- **Vary the look of the slides**
  - Some text-only slides are fine, but if there are too many in a row, audience falls asleep

That's all Folks!