Uncouth users, dopey developers, and crazy cryptographers

Why it's **never** the security architect's fault

OSS-NA 2026
Mike Bursell & CRob

Table of contents

01

Introductions

Wherein the audience meets our key players….

Who are these guys?

Mike

Executive Director, Confidential Computing Consortium

Author, ex-founder, geek, ex-CEO, single malt enthusiast, West Wing aficionado, "Abu el Banat"

CRob

OpenSSF CTO & Chief Security Architect

chmod 666 crob.md

44th level Dungeon Master

27th level Securityologist

We’re Architects

We’re both Security Architects

Image source

…but we’re not the same type of security architect!

We're Architects

Mike is a System Architect
(with a spicy dash of solution architecture)

CRob is an Operations Architect
(with a heaping helping of IAM and GRC)

Image source

We ***OBVIOUSLY*** hold each other in utter contempt (boo hiss)

But not as much as that we reserve for people who are ***NOT*** security architects (those *people* are JUST THE WORST!)

02

Architect?

Where the audience learns we do not draw buildings and bridges all day long

Or *DO* we?

“But I assumed that all architects were the same…”

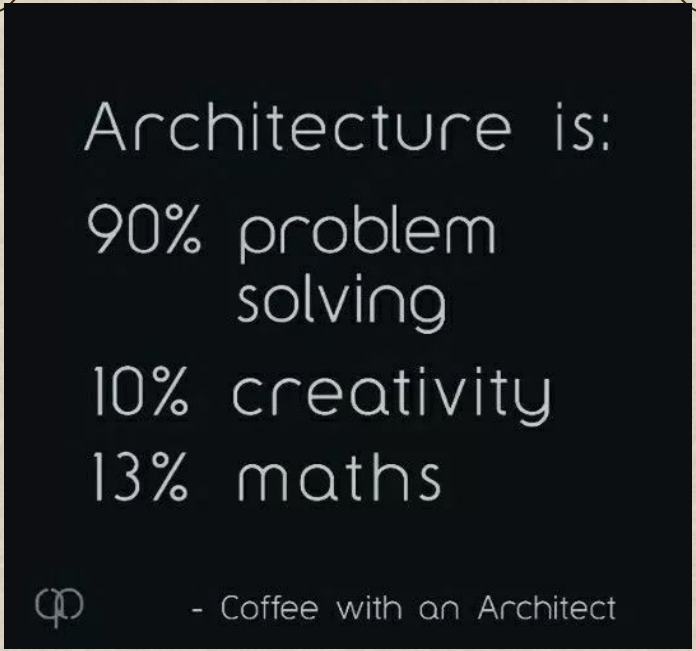


Image source

POPPYCOCK! What a foolish notion!

There are lots of different types of architects!

Application, solution, system, enterprise, operational … and even within those, there are differences

…and security specialisms across each

You need security EVERYWHERE!

Silos

Point solutions
Islands of identity & data
Duplicative tools

Siloing doesn't help

So many security frameworks to choose from!

TOGAF
UML views
SABSA
M1TRE
Zachman
UAF
WoW
LoTR

DON'T PANIC

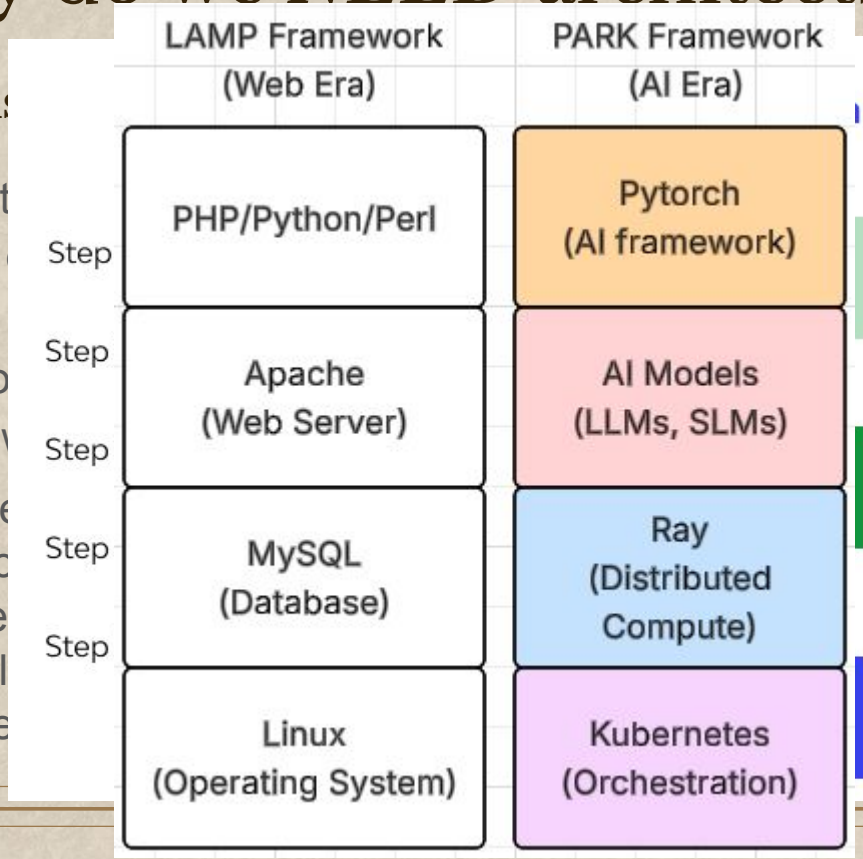
It's tricky, and that's why we're the cleverest, best educated and, let's face it, sexiest folks in IT.

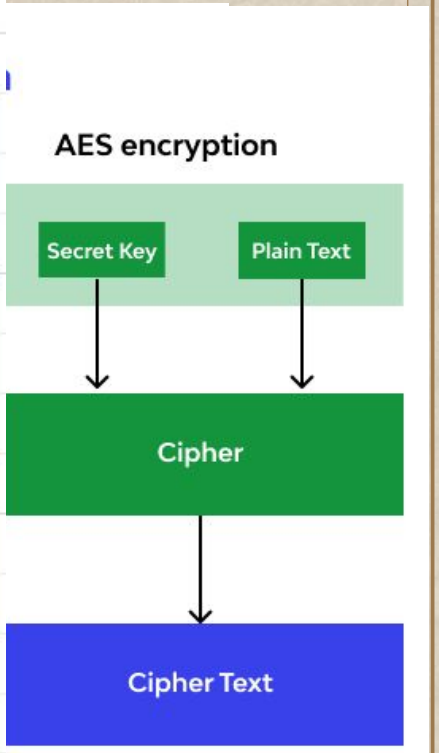
Image source

Why do we *NEED* architects?

Abstractions

- You can't know all t
  (and it was getting
  started)
- Most folks look at o
  (hopefully) know it
- But as you combine
  stages or applicatio
  emergent propertie
  only exist as compl
  That's why you nee





Getting the picture yet?

Get it? “Picture”? We amuse ourselves

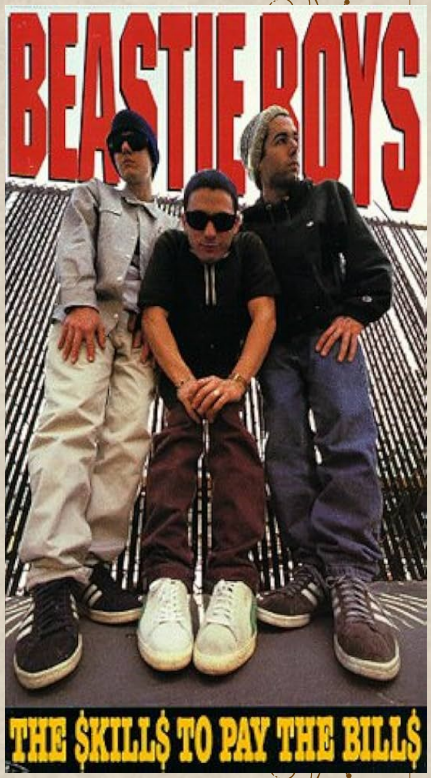
Architects: “we got skillz”

Operations are about business, resilience, risk and continuity. C-level folks don’t care about security (even CISOs) - they care about covering their …

… risk.

Most security folks are trained to start with the “*no*”. This doesn’t go down well with the C-levels.

Architects start with “*probably no, but pay us lots and we’ll find a maybe*”.



03

The interesting people we get to work with

Wherein we take the order from the Customer and give it to the Engineers!

We’ve got PEOPLE SKILLS!!!

A Taxonomy of selected personae

Who?	What do they do wrong?	What should they do?
Users	Stupidity	
Engineers	Don’t do security	
Security engineers	Only think about their domain	
Cryptographers		
Testers		
Ops folks		
Product managers		
Executives		
Security architects		

A Taxonomy of selected personae

Who?	What do they do wrong?	What should they do?
Users	Stupidity	
Engineers	Don’t do security	
Security engineers	Only think about their domain	
Cryptographers	Believe that mathematics fixes everything	
Testers	Assume that unit test pass = secure	
Ops folks	Believe that once it’s deployed, it’s good	
Product managers	Increase security technical debt	
Executives		
Security architects		

A Taxonomy of selected personae

Who?	What do they do wrong?	What should they do?
Users	Stupidity	
Engineers	Don’t do security	
Security engineers	Only think about their domain	
Cryptographers	Believe that mathematics fixes everything	
Testers	Assume that unit test pass = secure	
Ops folks	Believe that once it’s deployed, it’s good	
Product managers	Increase security technical debt	
Executives	Deprioritise security over sales	
Security architects	Absolutely nothing	

A Taxonomy of selected personae

Who?	What do they do wrong?	What should they do?
Users	Stupidity	Don’t use our code/RTFM
Engineers	Don’t do security	Learn to do security, at least in their domain
Security engineers	Only think about their domain	Realise that their code is part of a system
Cryptographers	Believe that mathematics fixes everything	Understand that (most) systems involve (fallible) people
Testers	Assume that unit test pass = secure	Look at how multiple components interact
Ops folks	Believe that once it’s deployed, it’s good	Think about how software has an attack lifecycle
Product managers	Increase security technical debt	Put security into every release
Executives	Deprioritise security over sales	Spend more money on security
Security architects	Absolutely nothing	Keep up the great work of understanding everything!!

An UPDATED Taxonomy of selected personae

Who?	What do they do wrong?	What should they do?
Users	Stupidity	Don’t use our code/RTFM
Engineers	Don’t do security	Talk to a security architect
Security engineers	Only think about their domain	Talk to a security architect
Cryptographers	Believe that mathematics fixes everything	Talk to a security architect
Testers	Assume that unit test pass = secure	Talk to a security architect
Ops folks	Believe that once it’s deployed, it’s good	Talk to a security architect
Product managers	Increase security technical debt	Talk to a security architect
Executives	Deprioritise security over sales	Spend more money on security
Security architects	Absolutely nothing	Keep up the great work of understanding everything!! <3

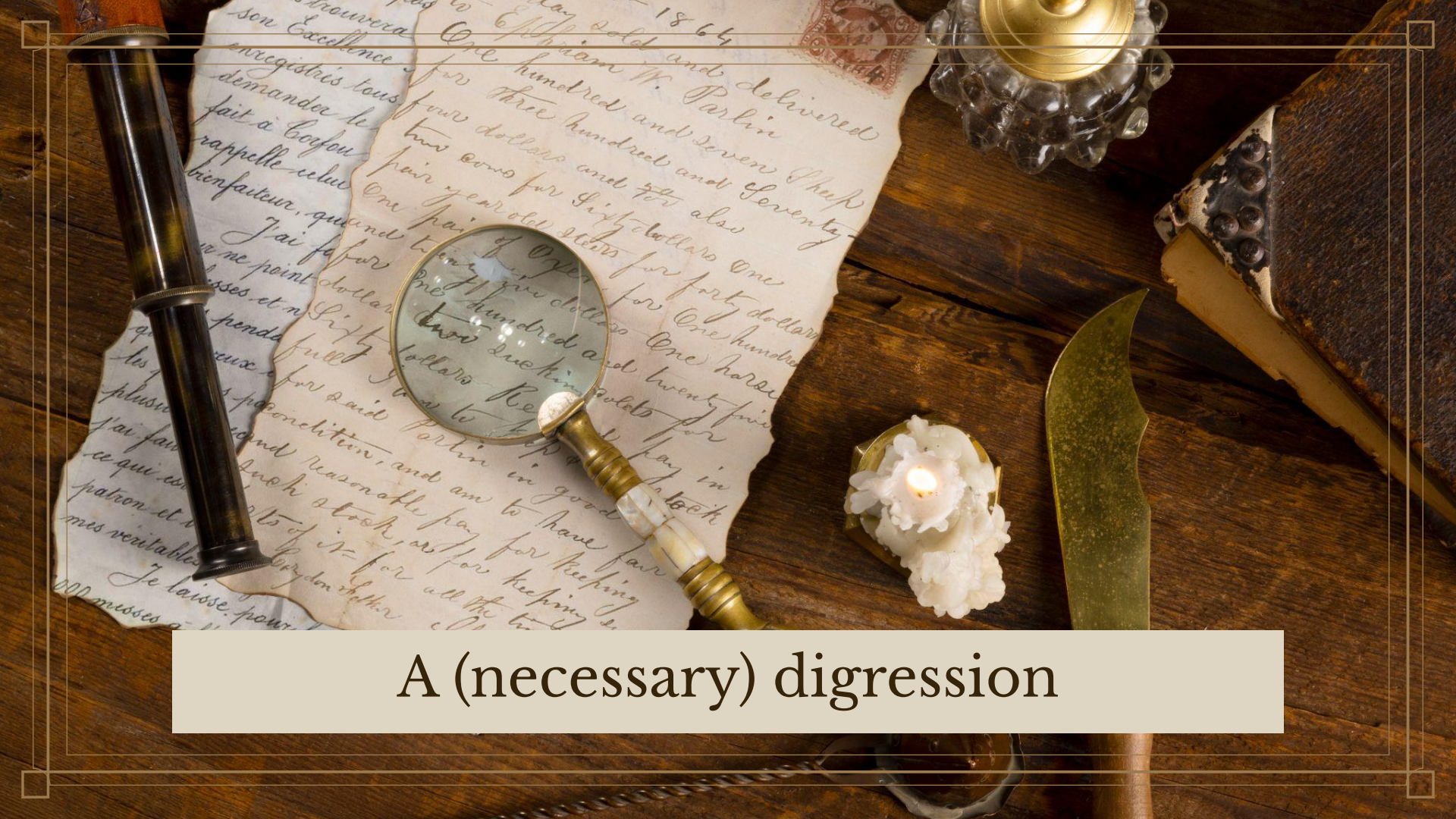
An **UPDATED** Taxonomy of selected personae

Who?	**What do they do wrong?**	**What should they do?**
Users	Stupidity	Don’t use our code/RTFM
Engineers	Don’t do security	Talk to a security architect
Security engineers	Only think about their domain	Talk to a security architect
Cryptographers	Believe that mathematics fixes everything	Talk to a security architect
Testers	Assume that unit test pass = secure	Talk to a security architect
Ops folks	Believe that once it’s deployed, it’s good	Talk to a security architect
Product managers	Increase security technical debt	Talk to a security architect
Executives	Deprioritise security over sales	Spend more money on security **architects**
Security architects	Absolutely nothing	Keep up the great work of understanding everything!! <3

04

The Unbearable Lightness of Being an Architect

Wherein our heroes explain what they ACTUALLY do

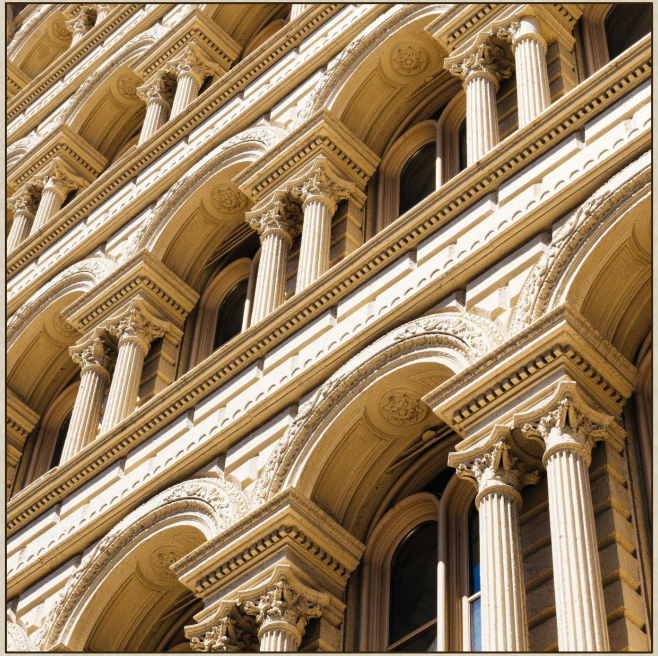


A (necessary) digression

Why open source is better than anything else

"Open source is the worst form of code, except for all the others" (Winston Churchill)

- If you can't see it, you can't fix it
- Proprietary == just your own experts
- Open source == a whole world of experts (if they're interested/incentivi[s|z]ed)
  - "All bugs are shallow" **only** counts if you have relevant people who actually check
- If no-one else fixes it, you can (but be careful!)

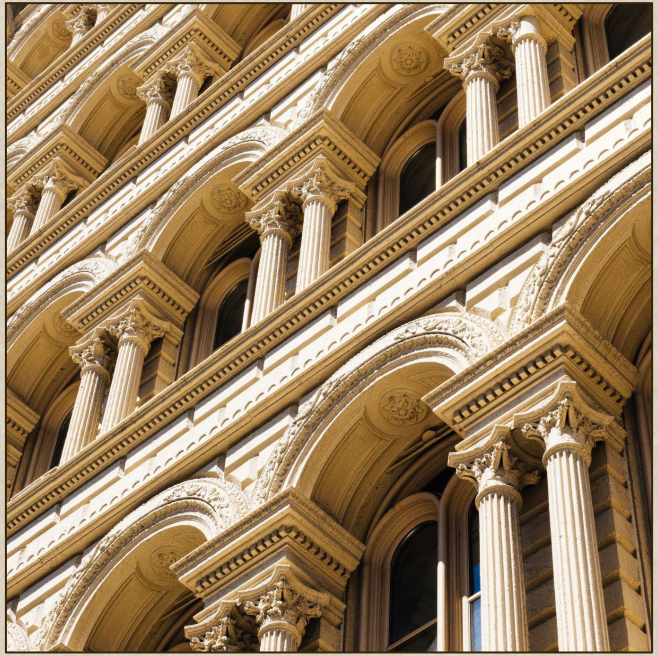


But now, more importantly…. Back to the Architects!

Why ~~open source~~ architects are better than ~~anything~~ everyone else

“Architects are the worst form of security folks, except for all the others” (Winston Churchill)

The wider context requires multiple views and viewpoints to see the wider picture and address different stakeholders’ requirements, allowing them to be able to take the right actions.



So what is it...you say you DO here?

Image source

- Circles and Boxes
- L33T Powerpointing
- Fancy Waistcoats
- Ivory Towers
- Brilliant Insights
- Scathing Wit

So what is it...you say you DO here?

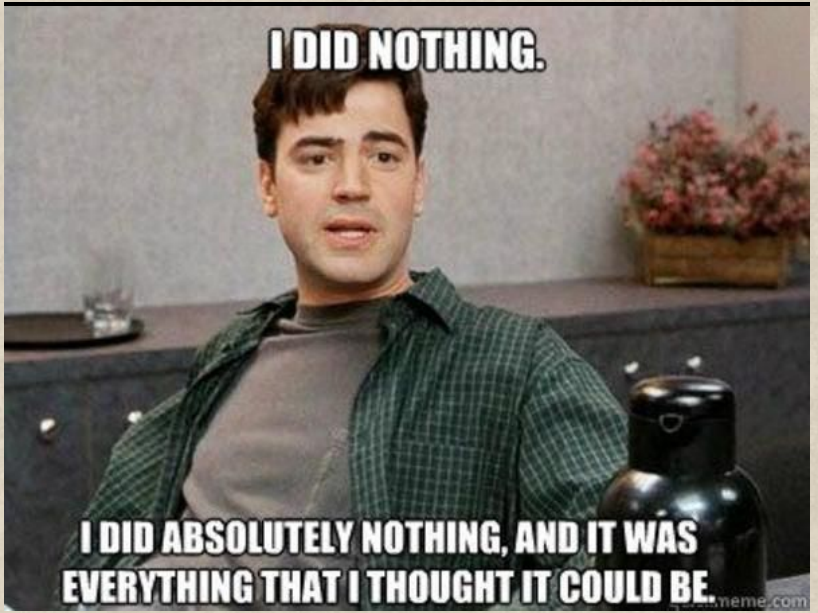
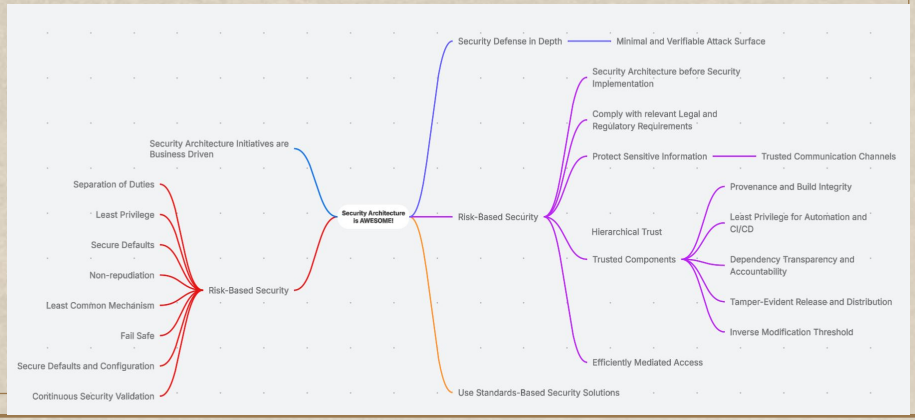


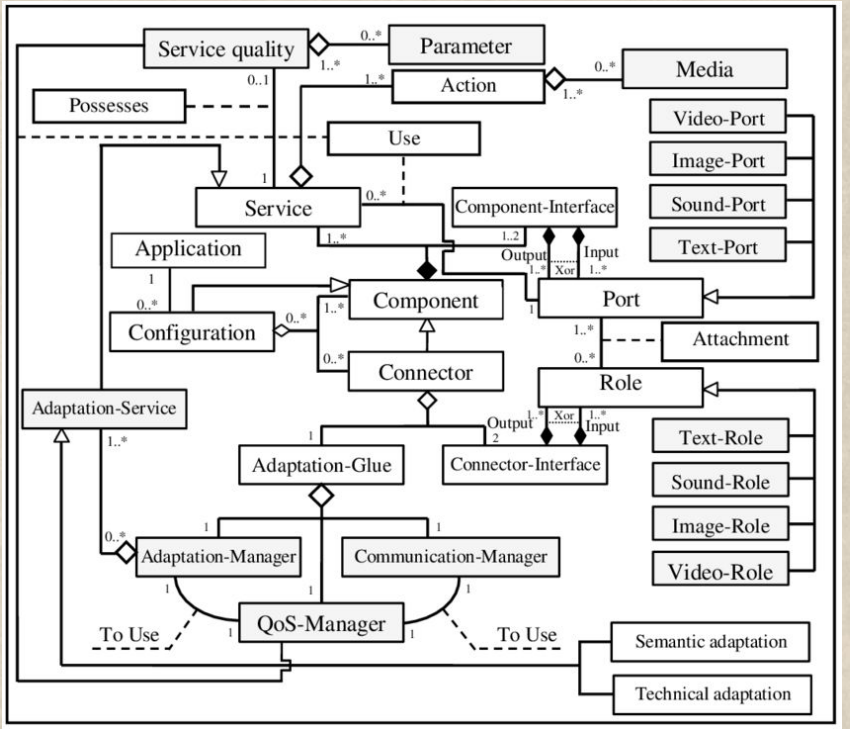
Image source

But seriously…

Understanding security principles and knowing how to apply them is a complex set of skills that takes time to learn



So what is it...you say you DO here?



And, to be fair, we *do* enjoy drawing boxes and lines.

05

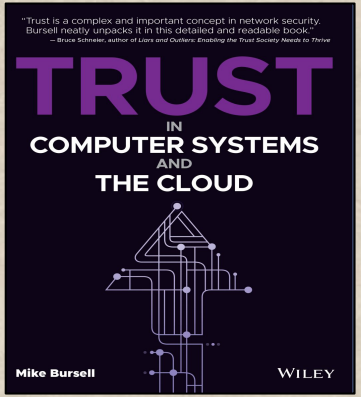
Conclusions?

Wherein our heroes get to the actual point

A few take-aways

CONFIDENTIAL COMPUTING CONSORTIUM

OpenSSF
OPEN SOURCE SECURITY FOUNDATION



...But Seriously(ish)

Think bigger…

...But Seriously(ish)

Think bigger…

…bridge communities

...But Seriously(ish)

Think bigger…

…bridge communities

…talk the language of your audience

...But Seriously(ish)

Think bigger…

…bridge communities

…talk the language of your audience

…see the bigger picture (and it keeps expanding)

...But Seriously(ish)

Think bigger…

…bridge communities

…talk the language of your audience

…see the bigger picture (and it keeps expanding)

… but be able to understand the lower level tech stuff (or able to fake it sufficiently)

...But Seriously(ish)

Think bigger…

~~…bridge communities~~

~~…talk the language of your audience~~

~~…see the bigger picture (and it keeps expanding)~~

~~… but be able to understand the lower level tech stuff (or able to fake it sufficiently)~~

… salaries for security architects

Thank you very much for your attention!

mike_at_p2pconsulting.dev

https://github.com/MikeCamel

What is CyberSecurity?

https://www.linkedin.com/in/mikebursell/

CRob_at_OpenSSF_dot_org

@~~SecurityCRob~~

@SecurityCRob@infosec.exchange

https://github.com/SecurityCRob

The Security Unhappy Hour,
Chips & Salsa
What's in the SOSS?

https://www.linkedin.com/in/darthcrob/

06

Resources

A Catalogue of Interesting Things to Have Languishing on Your Shelf to Both Gaze Upon and Perhaps Someday Read

Good Reads

Ross Anderson *Security Engineering: A Guide to Building Dependable Distributed Systems*

Eduardo B. Fernandez *Security Patterns in Practice*

Rick Howard *CyberSecurity First Principles*

Jerome Saltzer and Michael Schroeder *The Protection of Information in Computer Systems*

Brook S. E. Schoenfield *Secrets of a Cyber Security Architect*

Brook S. E. Schoenfield *Securing Systems - Applied Security Architecture and Threat Models*

John Sherwood, Andrew Clark, and David Lymas *Enterprise Security Architecture*

Diomedis Spinellis and Georgious Gousios *Beautiful Architecture: Leading Thinkers Reveal the Hidden Beauty in Software Design*

Stephen Wahe *Open Enterprise Security Architecture*