

---

---

# A Pictorial Introduction to Components in Scientific Computing

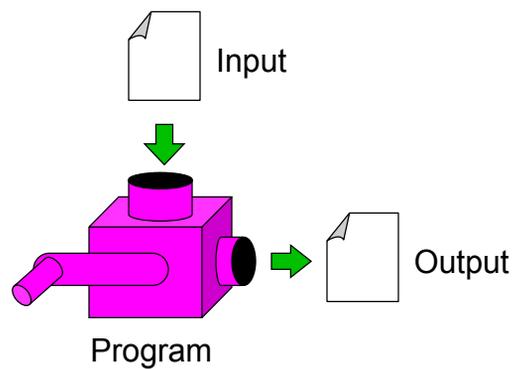
---

---

Once upon a time...

---

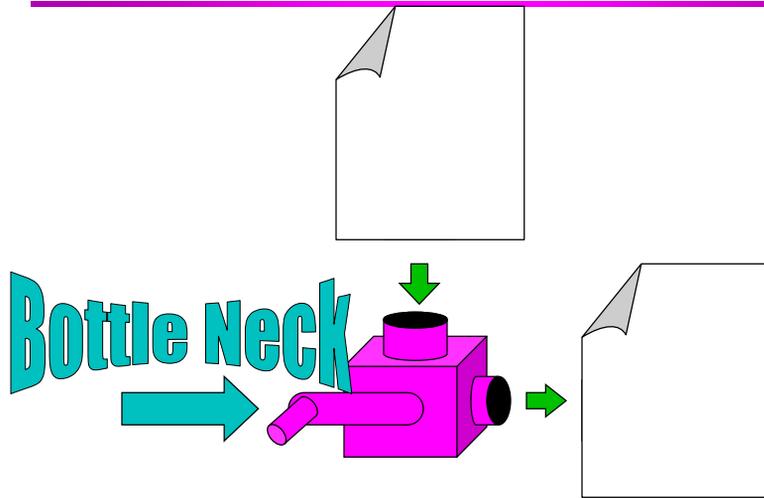
---



## As Scientific Computing grew...

---

---

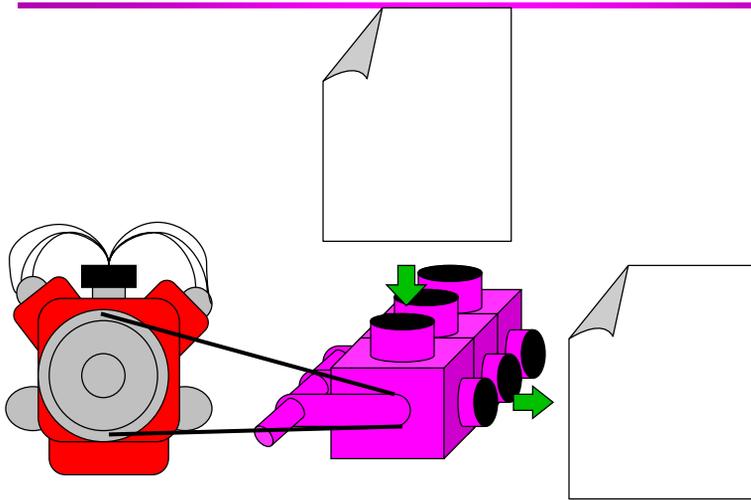


3

## Tried to ease the bottle neck

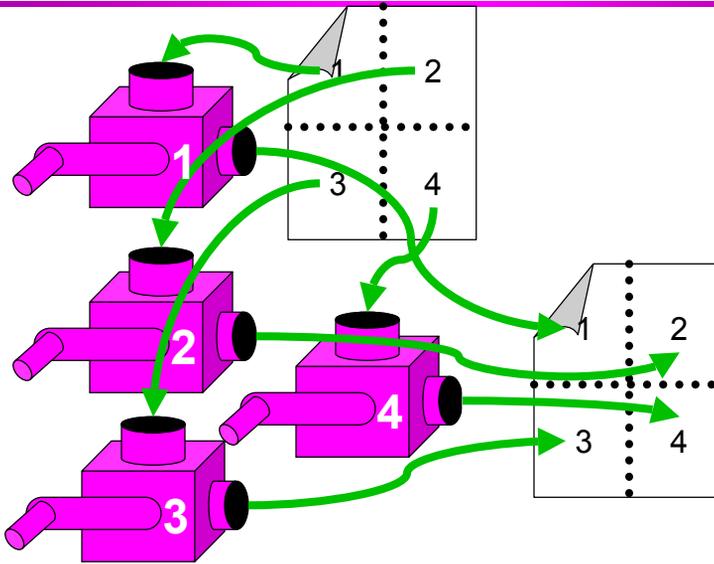
---

---

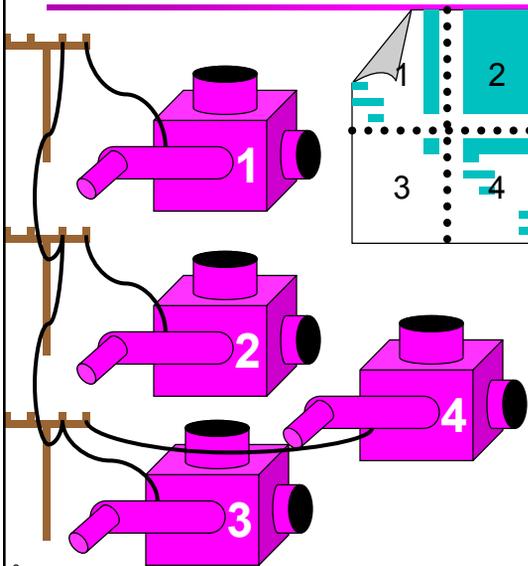


4

# SPMD was born.

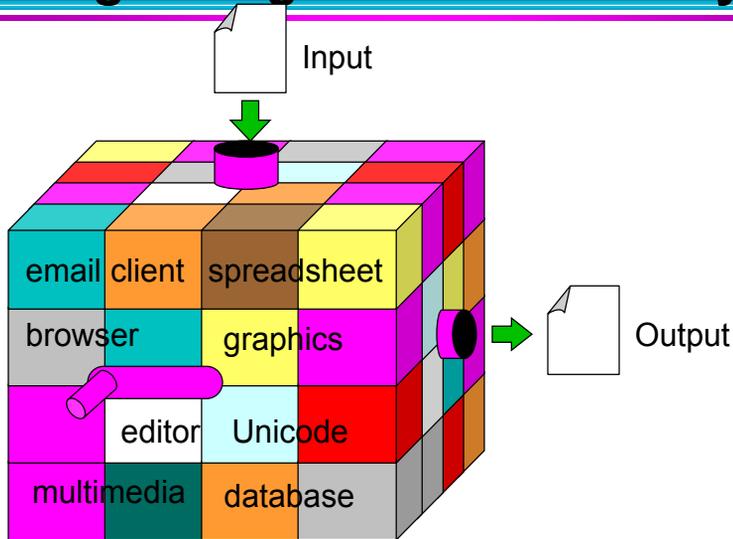


# SPMD worked.



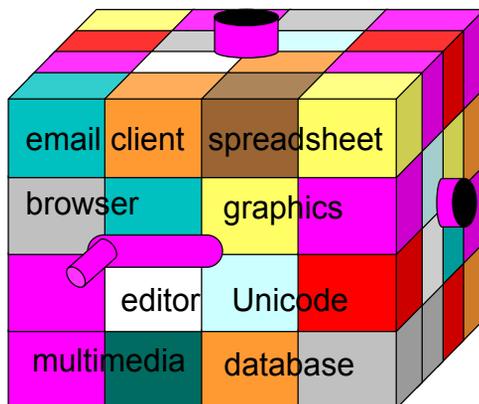
**But it  
isn't  
easy!!!**

## Meanwhile, corporate computing was growing in a different way



7

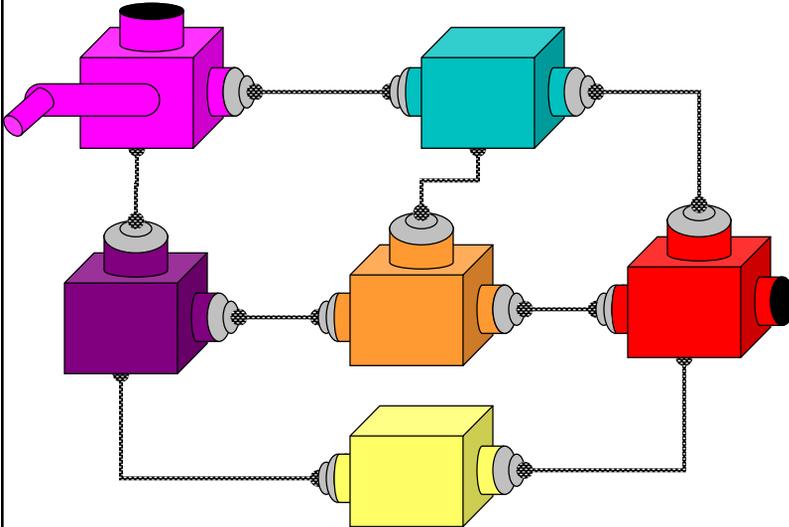
## This created a whole new set of problems → complexity



- Interoperability across multiple languages
- Interoperability across multiple platforms
- Incremental evolution of large legacy systems (esp. w/ multiple 3rd party software)

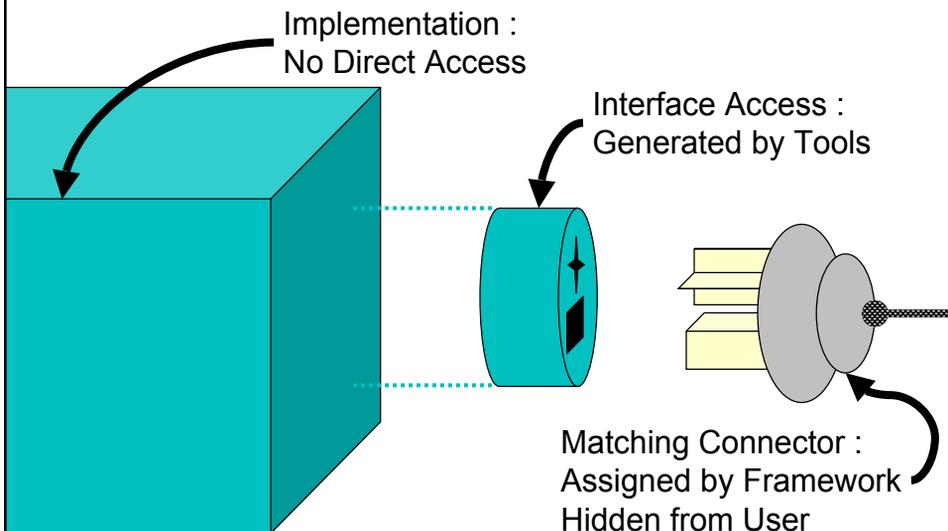
8

# Component Technology addresses these problems



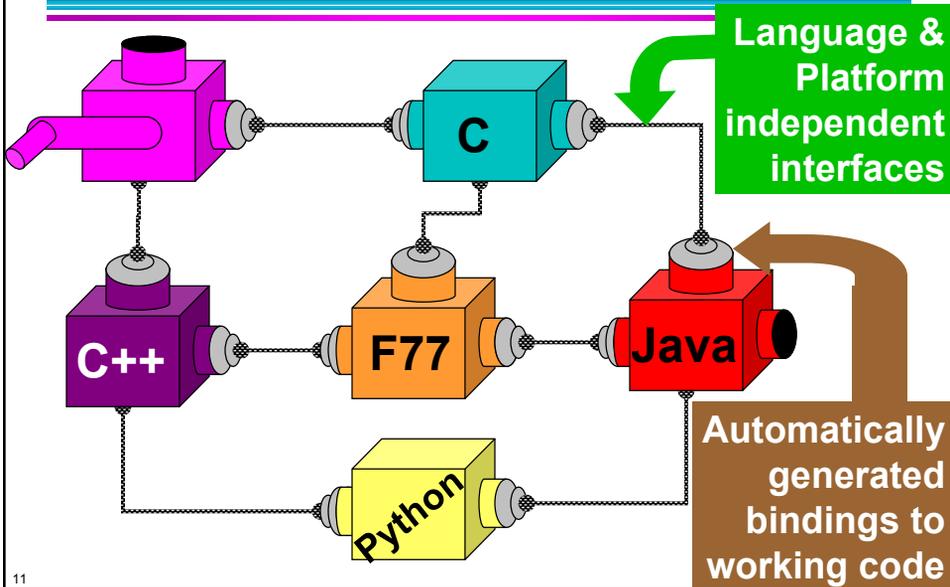
9

# So what's a component ???

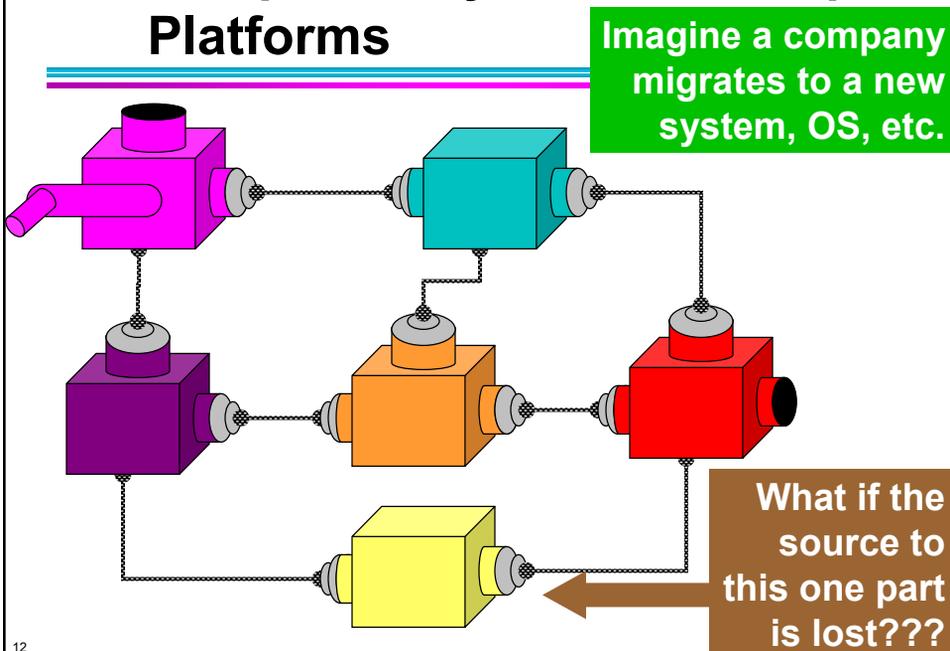


10

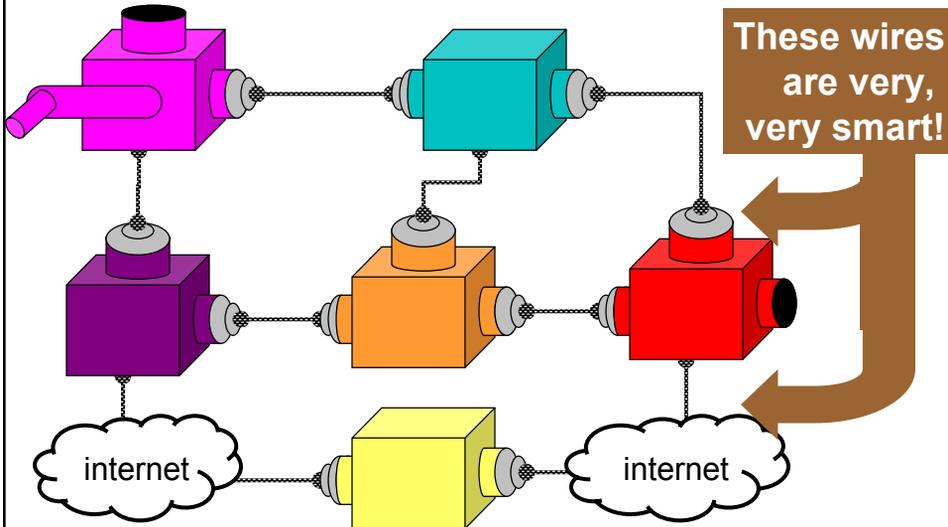
# 1. Interoperability across multiple languages



# 2. Interoperability Across Multiple Platforms

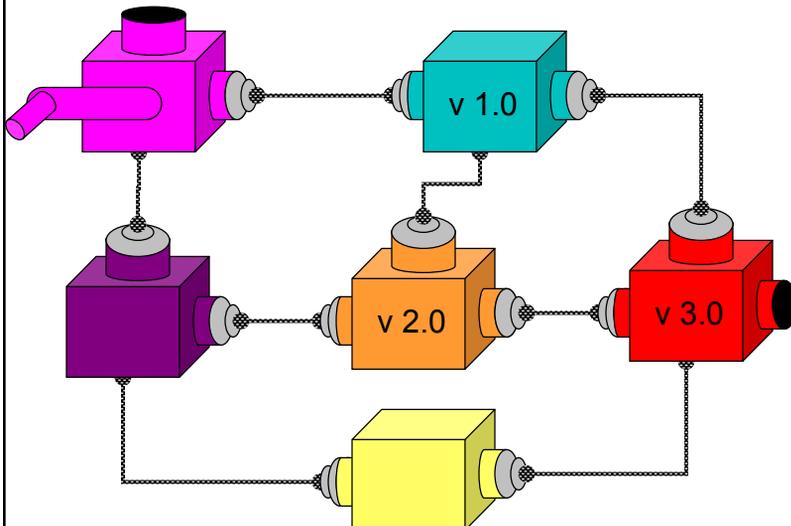


# Transparent Distributed Computing



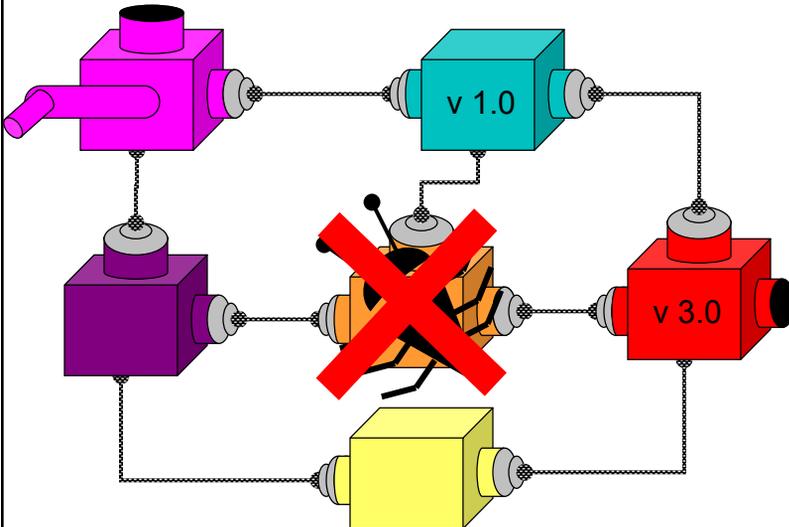
13

## 3. Incremental Evolution With Multiple 3rd party software



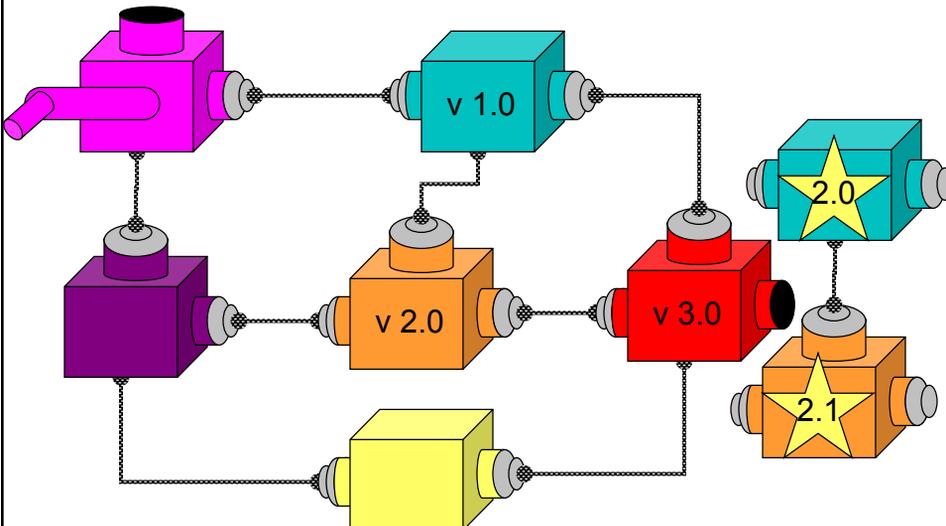
14

**Now suppose you find this bug...**



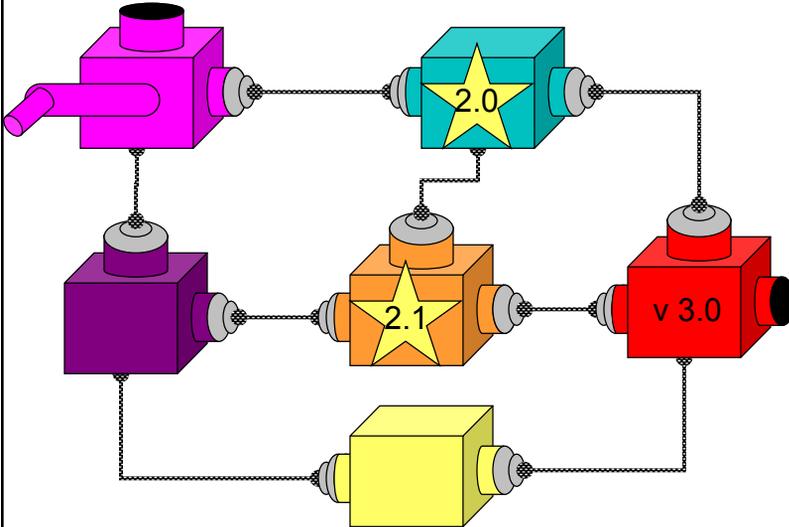
15

**Good news: an upgrade available**  
**Bad news: there's a dependency**



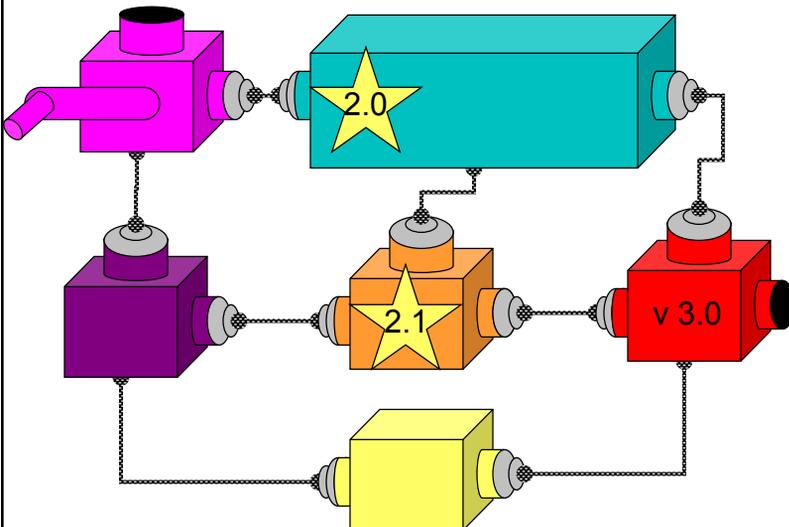
16

# Great News: Solvable with Components



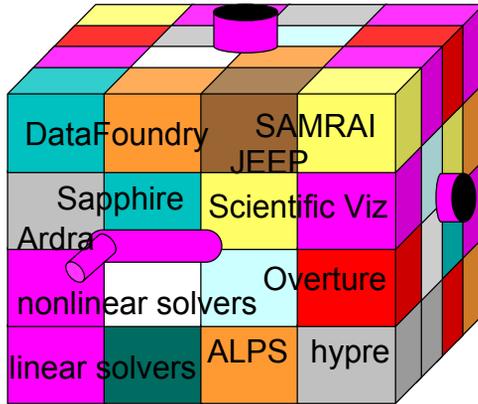
17

# Great News: Solvable with Components



18

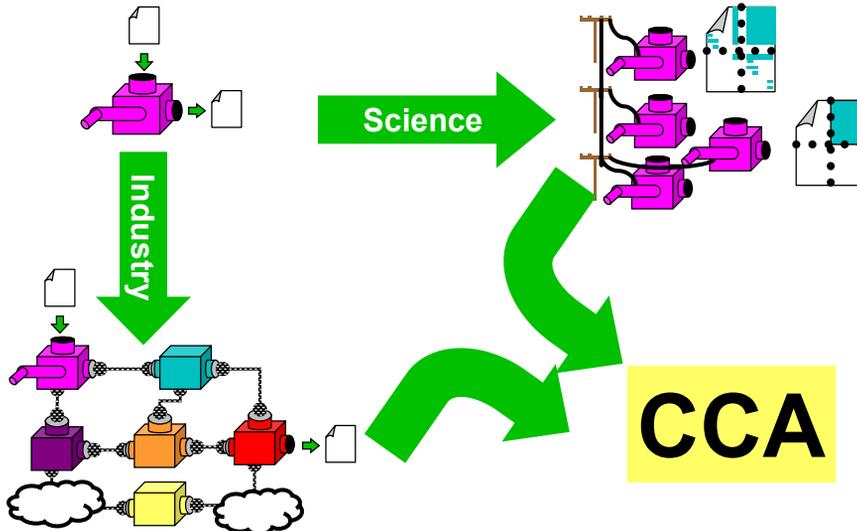
# Why Components for Scientific Computing → Complexity



- Interoperability across multiple languages
- Interoperability across multiple platforms
- Incremental evolution of large legacy systems (esp. w/ multiple 3rd party software)

19

# The Model for Scientific Component Programming



20



# The End

Next: [Intro to Components](#)