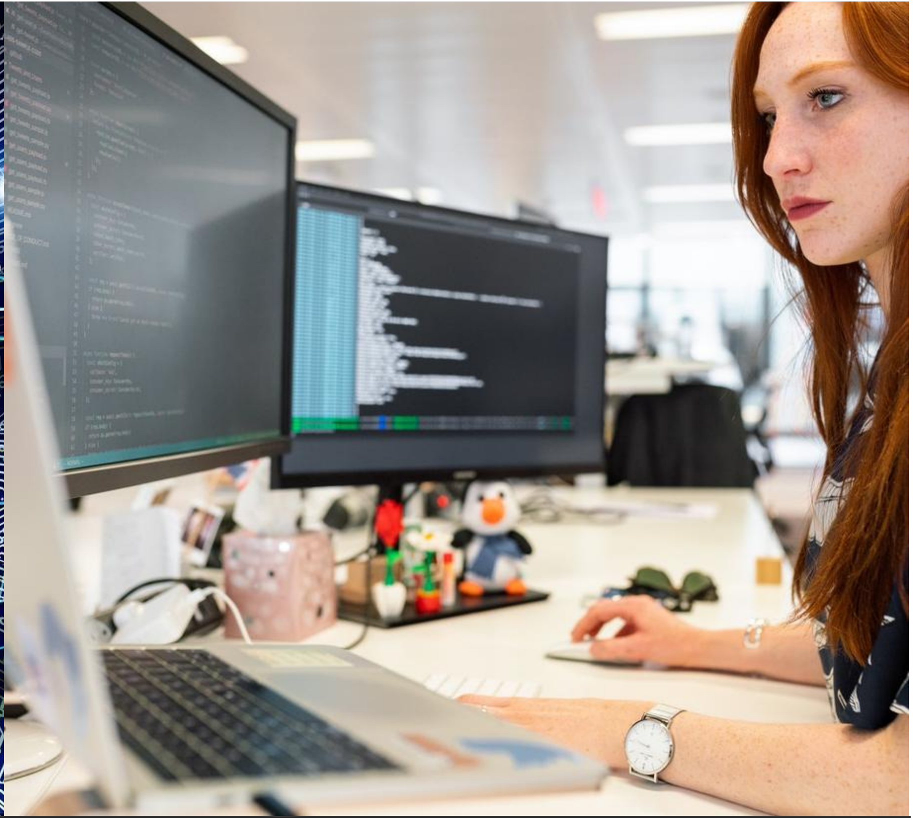
Software Engineering for Artificial Intelligence

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Outro



That‘s it…

This seminar heavily relied on your work...

...thanks for doing it well!

But wait!

This journey can continue: We are looking for great people to do research with!

We offer:

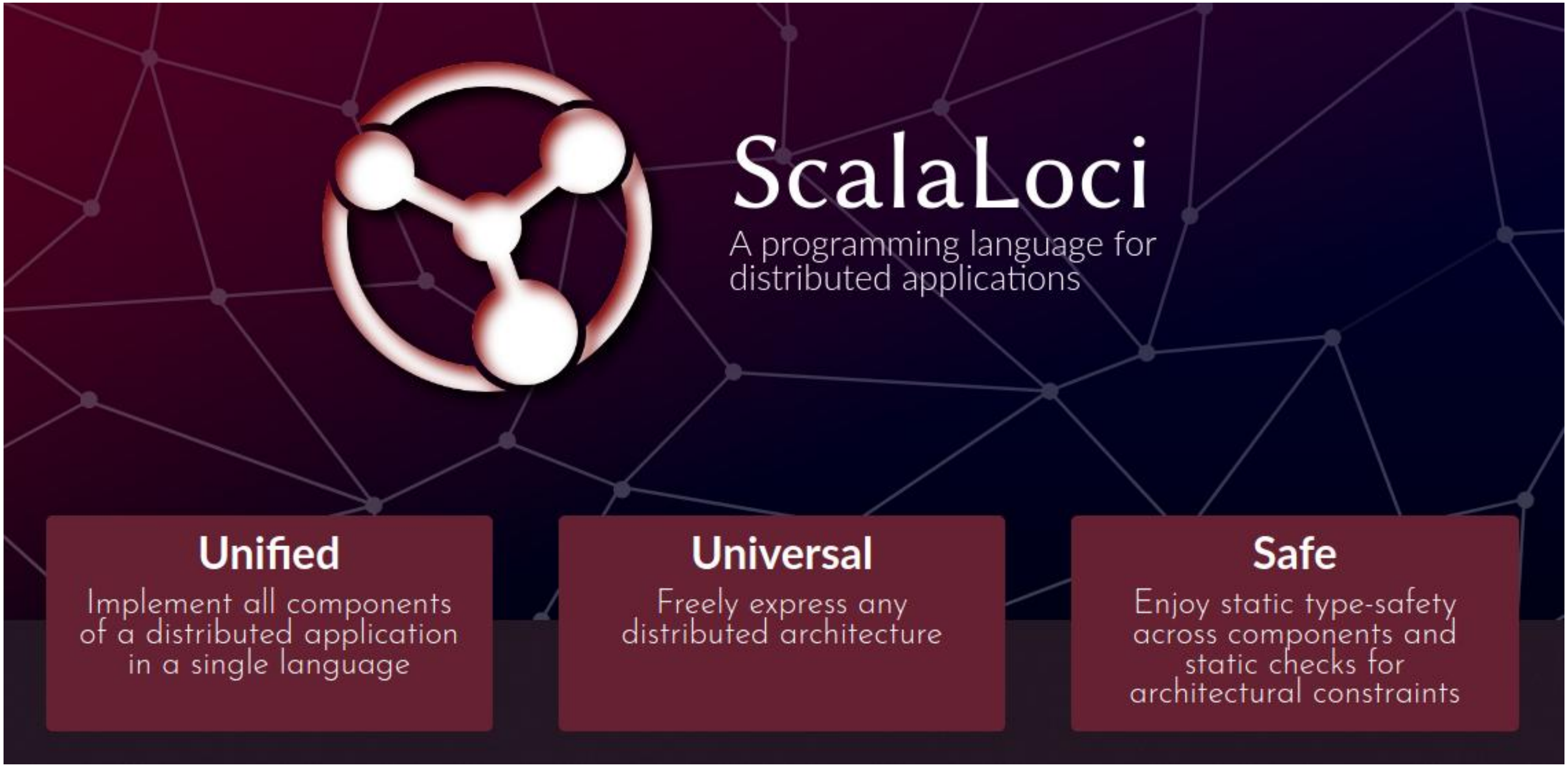
- Thesis (Bachelor & Master)
- Research Seminar and Projects (IMPL & DAIMPL)
- HiWi Jobs (Bachelor & Master)
- PhD Positions (post Master)

What do we do?

We solve problems using Programming Language Techniques,

especially, in Software Engineering and Distributed Systems.

Multitier Programming with ScalaLoci



ScalaLoci

A programming language for distributed applications

Unified

Implement all components of a distributed application in a single language

Universal

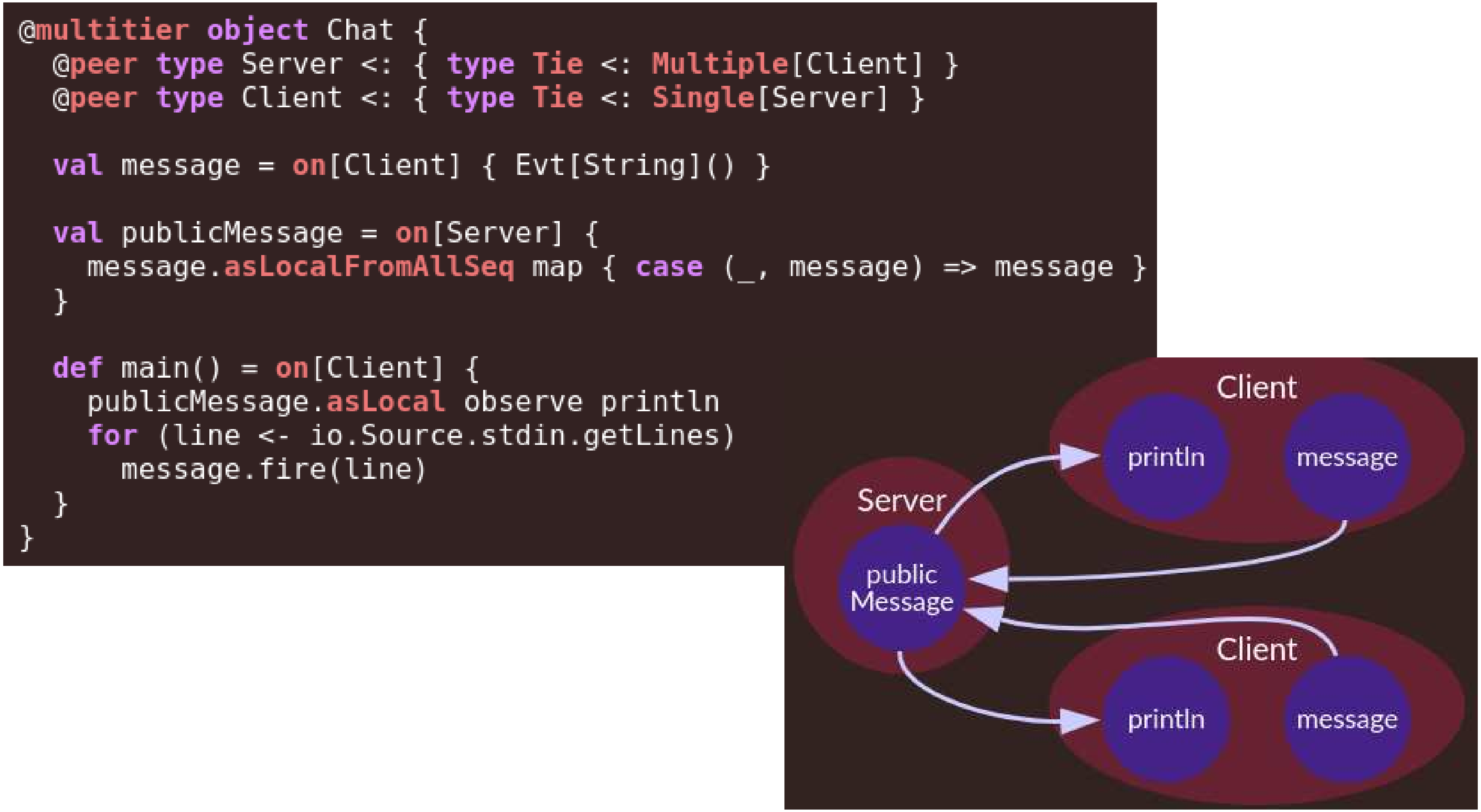
Freely express any distributed architecture

Safe

Enjoy static type-safety across components and static checks for architectural constraints

https://scala-loci.github.io/

ScalaLoci Example



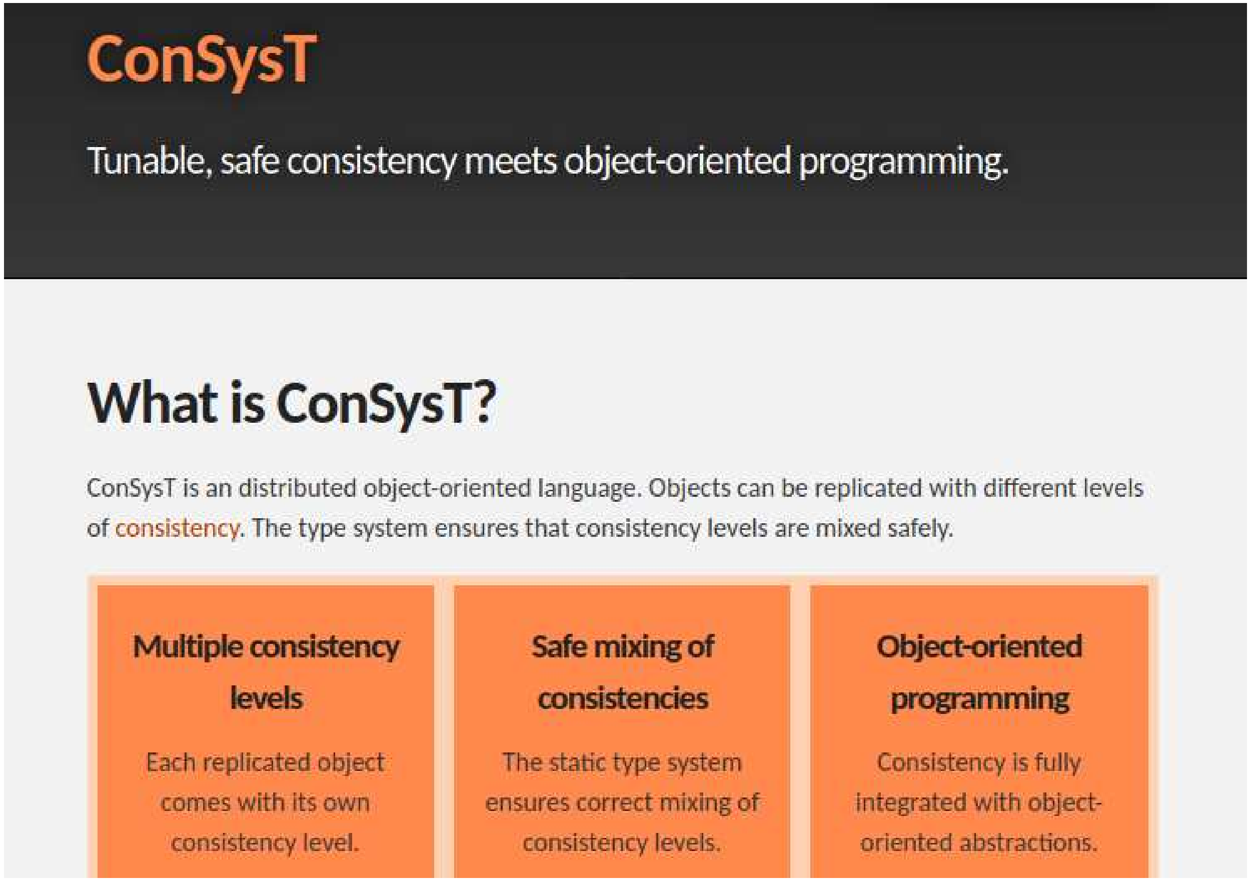
```
@multitier object Chat {
  @peer type Server <: { type Tie <: Multiple[Client] }
  @peer type Client <: { type Tie <: Single[Server] }

  val message = on[Client] { Evt[String]() }

  val publicMessage = on[Server] {
    message.asLocalFromAllSeq map { case (_, message) => message }
  }

  def main() = on[Client] {
    publicMessage.asLocal observe println
    for (line <- io.Source.stdin.getLines)
      message.fire(line)
  }
}
```

Mixing Consistency in OOP with ConSysT



ConSysT

Tunable, safe consistency meets object-oriented programming.

What is ConSysT?

ConSysT is an distributed object-oriented language. Objects can be replicated with different levels of consistency. The type system ensures that consistency levels are mixed safely.

Multiple consistency levels

Each replicated object comes with its own consistency level.

Safe mixing of consistencies

The static type system ensures correct mixing of consistency levels.

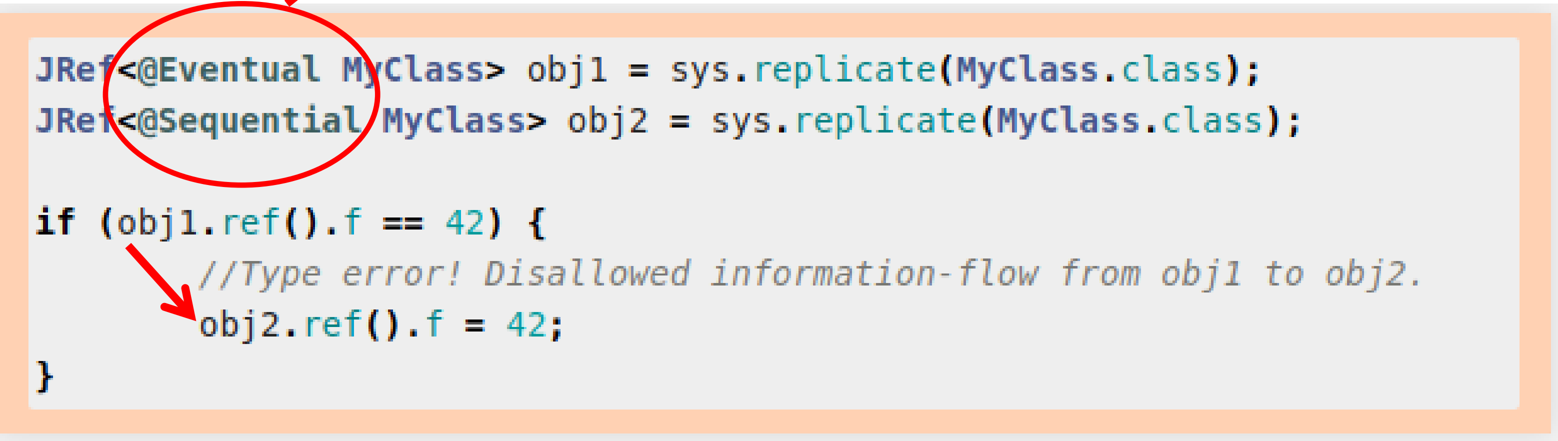
Object-oriented programming

Consistency is fully integrated with object-oriented abstractions.

https://consyst-project.github.io/

ConSysT Example

Distributed objects with consistency levels



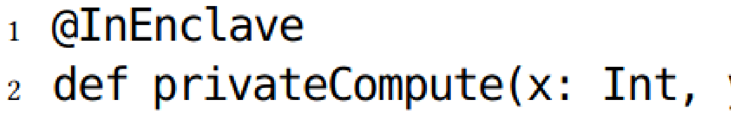
```
JRef<@Eventual MyClass> obj1 = sys.replicate(MyClass.class);
JRef<@Sequential MyClass> obj2 = sys.replicate(MyClass.class);

if (obj1.ref().f == 42) {
    //Type error! Disallowed information-flow from obj1 to obj2.
    obj2.ref().f = 42;
}
```

Stronger (@Sequential) consistency values must not depend on values of weaker consistency (@Eventual)!

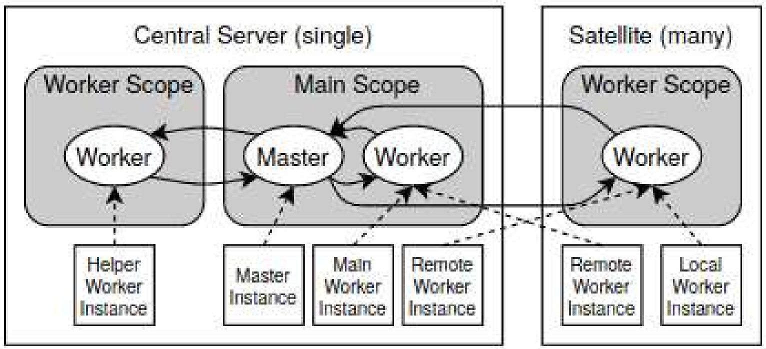
Projects in earlier stages

- Language abstracts for privacy enforcing techniques, like Intel SGX, homomorphic encryption, etc.



```
@InEnclave
def privateCompute(x: Int,
```

- Decentralized orchestration and dynamic placement for distributed systems
  - Language to define the deployment of components decentrally, enforcing inter-component dependency availability and supporting movement of components



- Consistency and language abstractions in serverless computing

How does SE4AI relate to that?

- We want to add AI to our problem focus:
  - How can we improve AI development using Programming Language Techniques?
    - E.g., using PL techniques for AI testing, safer model specification languages, automated verification of AI systems
  - How can our current projects leverage ML?
    - E.g., ML-driven placement decisions, composing AI models with multitier programming, safe mixing of accurracy levels

Contact

Prof. Guido Salvaneschi
salvaneschi@cs.tu-darmstadt.de

You liked one of our projects or have a great new idea to start with?

Join us!

Acknowledgements & License

- Images are either by the authors of these slides, attributed where they are used, or licensed under Pixabay or Pexels
- These slides are made available by the authors (Daniel Sokolowski, Guido Salvaneschi) under CC BY 4.0