# Collaboration Notes: Prof. Joost Visser

**Professor - Large Scale Software & Data Science**

## Background Research

### Key Focus Areas

1. **Software Engineering for Machine Learning (SE4ML)**
   - Leader of SE4ML project (https://se-ml.github.io/)
   - Focus: How to engineer software with ML components
   - Key insight: ML brings unique challenges to traditional SE
2. **Software Quality & Metrics**
   - Former CTO at Software Improvement Group (SIG)
   - Created standardized maintainability models
   - Co-authored O'Reilly book "Building Maintainable Software"
   - Over 9,500 citations
3. **Current Research Themes**
   - ML trustworthiness and robustness
   - AutoML adoption in practice
   - Software architecture for ML systems
   - Green software and energy efficiency

### Academic Leadership

- Program Manager: Master in ICT in Business and the Public Sector
- Head of LIACS Software Lab
- Former part-time professor at Radboud University

## Understanding SE4ML

### The Challenge

Traditional software engineering assumes deterministic behavior, but ML components are:

- **Probabilistic**: Outputs can vary
- **Data-dependent**: Behavior changes with training data
- **Opaque**: Hard to understand decisions
- **Evolving**: Models need retraining

### Key Research Questions

1. How to test ML-based systems?
2. How to ensure robustness against adversarial inputs?
3. How to architect systems with ML components?
4. What practices do successful ML teams use?

### His Approach

- **Empirical studies**: Survey real ML practitioners
- **Best practices**: Identify what works in industry
- **Tool support**: Develop techniques for ML quality
- **Metrics**: Measure ML system quality

## Concrete Collaboration Ideas

### Project 1: API Misuse in ML Systems

**Goal**: Study and prevent API misuse patterns specific to ML frameworks

**Technical Approach:**

1. **Empirical Study**:
   - Analyze TensorFlow/PyTorch API misuse on GitHub
   - Identify patterns unique to ML development
   - Compare with traditional API misuse
2. **jGuard for ML APIs**:
   - Annotate common ML APIs (data loading, model training)
   - Guards for ML-specific constraints (tensor shapes, device placement)
   - Meta-variables for different hardware contexts (CPU/GPU/TPU)
3. **Metrics & Evaluation**:
   - Define ML-specific API quality metrics
   - Measure impact on model performance
   - Study developer productivity gains

**Why This Matters:**

- ML APIs are notoriously error-prone
- Mistakes can be subtle (wrong tensor dimensions)
- His SE4ML expertise + your API safety = novel contribution

### Project 2: Trustworthy ML Through API Contracts

**Goal**: Ensure ML robustness via API-level enforcement

**Technical Approach:**

1. **Robustness Contracts**:
   - Express adversarial robustness as API contracts
   - Guards ensure inputs within expected distribution
   - Consequences track model confidence
2. **Architecture Patterns**:
   - Study his work on ML architectures
   - Identify where API contracts add value
   - Design patterns for robust ML systems
3. **Tool Integration**:
   - Connect jGuard with ML testing tools
   - Automated contract generation from datasets
   - Integration with MLOps pipelines

**Connection to His Work:**

- Builds on his adversarial robustness research
- Applies his architectural insights
- Addresses trustworthy ML goals

### Project 3: Green ML Through Efficient APIs

**Goal**: Reduce ML energy consumption via better API usage

**Technical Approach:**

1. **Energy-Aware Guards**:
   - Monitor resource usage at API level
   - Prevent inefficient patterns (unnecessary GPU transfers)
   - Optimize batch sizes dynamically
2. **Measurement Framework**:
   - Extend his software metrics to energy
   - Profile ML API usage patterns
   - Identify energy hotspots
3. **Best Practices**:

  - Document energy-efficient API usage
  - Create jGuard rules for green ML
  - Industry case studies

**Benefits:**

- Addresses sustainability (hot topic)
- Combines his metrics expertise with your API work
- Practical impact for ML practitioners

## How to Present Your Ideas

### Opening:

"I've been following your SE4ML work and see how API misuse is a critical but understudied aspect of ML system quality..."

### Key Points to Emphasize:

1. **Empirical Foundation**: Show interest in studying real-world practices
2. **Practical Impact**: Focus on helping ML practitioners
3. **Metrics & Measurement**: Align with his quantitative approach
4. **Industry Relevance**: Mention SIG background appreciation

### Technical Terms to Use:

- **"Technical debt"**: ML systems accumulate unique forms
- **"Model drift"**: When deployed models degrade
- **"Pipeline debt"**: Complex ML workflows
- **"Architectural smells"**: Poor ML system design

## Questions to Ask Him

1. "What are the most critical API-related challenges you've seen in ML systems?"
2. "How do you see the role of contracts in ensuring ML trustworthiness?"
3. "What metrics would best capture ML API quality?"
4. "How can we bridge the gap between SE and ML communities?"

## Potential Joint Activities

### Papers:

1. **"An Empirical Study of API Misuse in Machine Learning Systems"**

    - Venue: ICSE or FSE
    - Mining study + developer survey

2. **"jGuard-ML: Contract-Based ML API Safety"**
    - Venue: ICSE-SEIP (Software Engineering in Practice)
    - Tool paper with industry evaluation
3. **"Green ML Through API Design: A Measurement Study"**
    - Venue: SANER or MSR
    - Energy analysis of ML APIs

### Grants:

- **NWO Applied Sciences**: Practical ML tools
- **EU Horizon Europe**: Trustworthy AI call
- **Industry collaboration**: Through his SIG connections

### Educational:

- Guest lectures in his Master program
- Student projects on ML API analysis
- Industry workshops on ML best practices

## Understanding His Network

### Academic Connections:

- **Arie van Deursen** (TU Delft): Software engineering
- **Erik Poll** (Radboud): Security expert
- **Alex Serban**: His PhD student on ML practices
- **Software Improvement Group**: Industry network

### Industry Perspective:

- Strong focus on practical applicability
- Values tools that work at scale
- Interested in tech transfer to industry

## Strategic Advantages

### For You:

1. **Industry validation**: His SIG connections
2. **Empirical expertise**: Strengthen your evaluation

3. **ML domain**: Expand jGuard to hot area
4. **Publication venues**: Access to top SE conferences

### For Him:

1. **Novel technique**: jGuard adds to SE4ML toolkit
2. **API perspective**: Understudied in ML systems
3. **Concrete tool**: Not just another survey
4. **PhD supervision**: Joint students possible

## Preparation Tips

### Before Meeting:

1. Read his ESEM 2020 paper on ML best practices
2. Check SE4ML website for current projects
3. Prepare examples of ML API misuse
4. Think about scalability (his focus on "large scale")

### During Meeting:

- Show empirical mindset (not just theory)
- Demonstrate industry awareness
- Be specific about ML challenges
- Mention teaching interests (he's program manager)

### Key Message:

"API contracts are a missing piece in the SE4ML puzzle - they can enforce the best practices your empirical work identifies"

## Recent Publications to Discuss

1. **"Adoption and Effects of Software Engineering Best Practices in Machine Learning"** (ESEM 2020)
   - Survey of ML practitioners
   - Identified gap between knowledge and practice
2. **"Adversarial Examples on Object Recognition"** (ACM Computing Surveys)
   - Comprehensive survey
   - Shows need for robustness techniques
3. **"Adapting Software Architectures to Machine Learning Challenges"** (SANER 2022)
   - Architectural patterns for ML

    - Where API contracts could help

## Cultural Fit

### His Style:

- Empirical and practical
- Industry-focused
- Metrics-driven
- Collaborative

### How to Align:

- Emphasize real-world impact
- Show measurement plans
- Discuss industry adoption
- Be open to large-scale studies

## Final Strategic Points

- He bridges academia-industry gap (valuable for career)
- SE4ML is growing field (good for funding)
- His students get industry connections
- Focus on "what works" not just "what's novel"
- Software Lab provides infrastructure for experiments