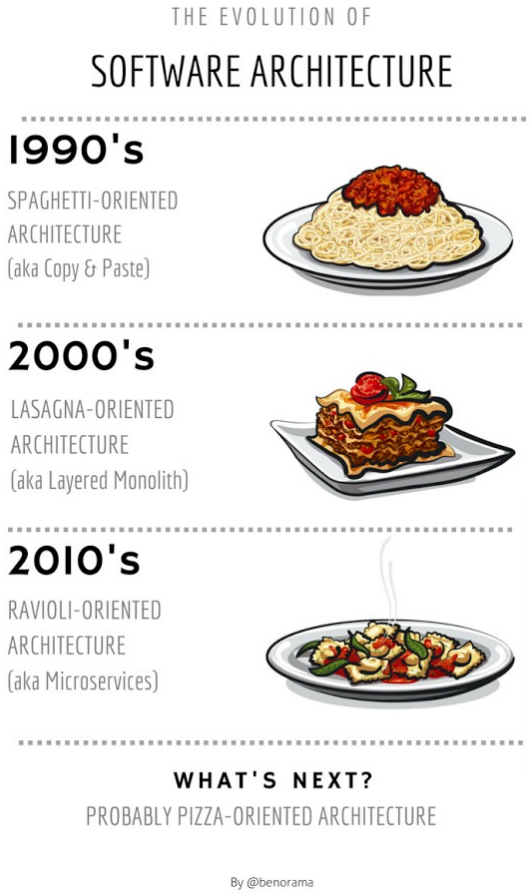
KU LEUVEN

The Odyssey: challenges to model privacy threats in a brave new world

Rafa Gálvez and Seda Gürses

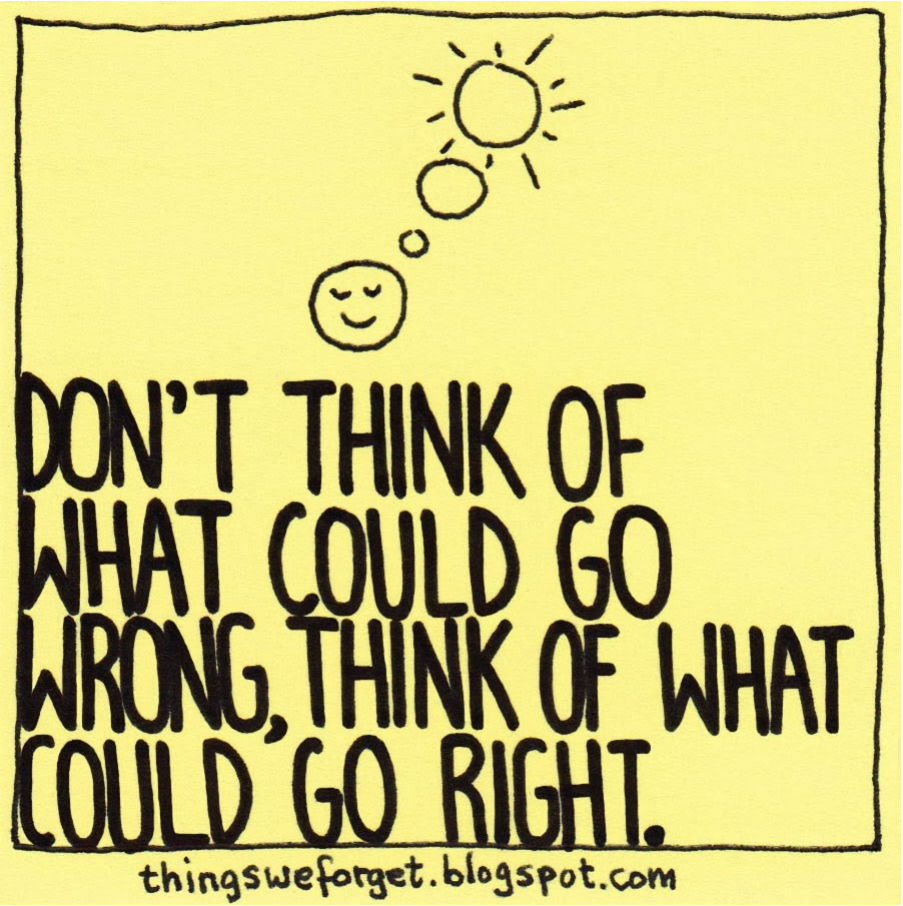
Motivation


GDPR


THE EVOLUTION OF
SOFTWARE ARCHITECTURE
1990's
SPAGHETTI-ORIENTED ARCHITECTURE
(aka Copy & Paste)
2000's
LASAGNA-ORIENTED ARCHITECTURE
(aka Layered Monolith)
2010's
RAVIOLI-ORIENTED ARCHITECTURE
(aka Microservices)
WHAT'S NEXT?
PROBABLY PIZZA-ORIENTED ARCHITECTURE
By @benorama

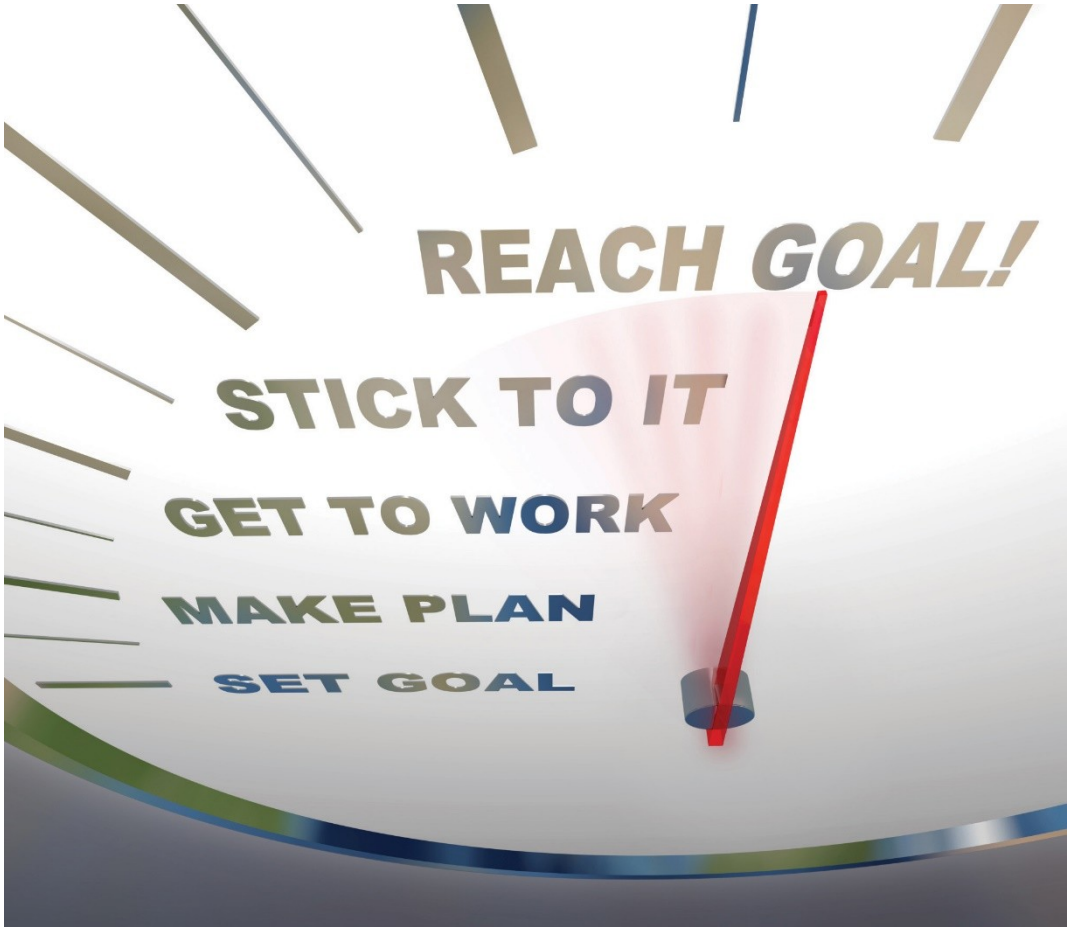
Threat Modeling

1. Characterize the system
2. Identify the threats
3. Threat and Risk analysis
4. Validate



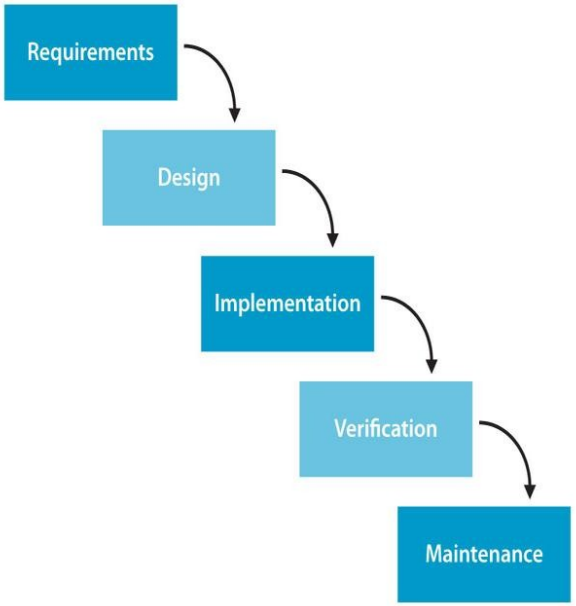
Privacy goals

- Confidentiality
- Control
- Practice

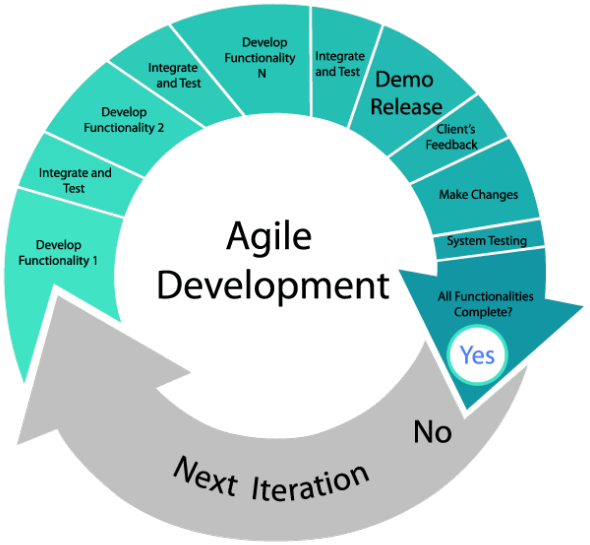


From waterfall to agile

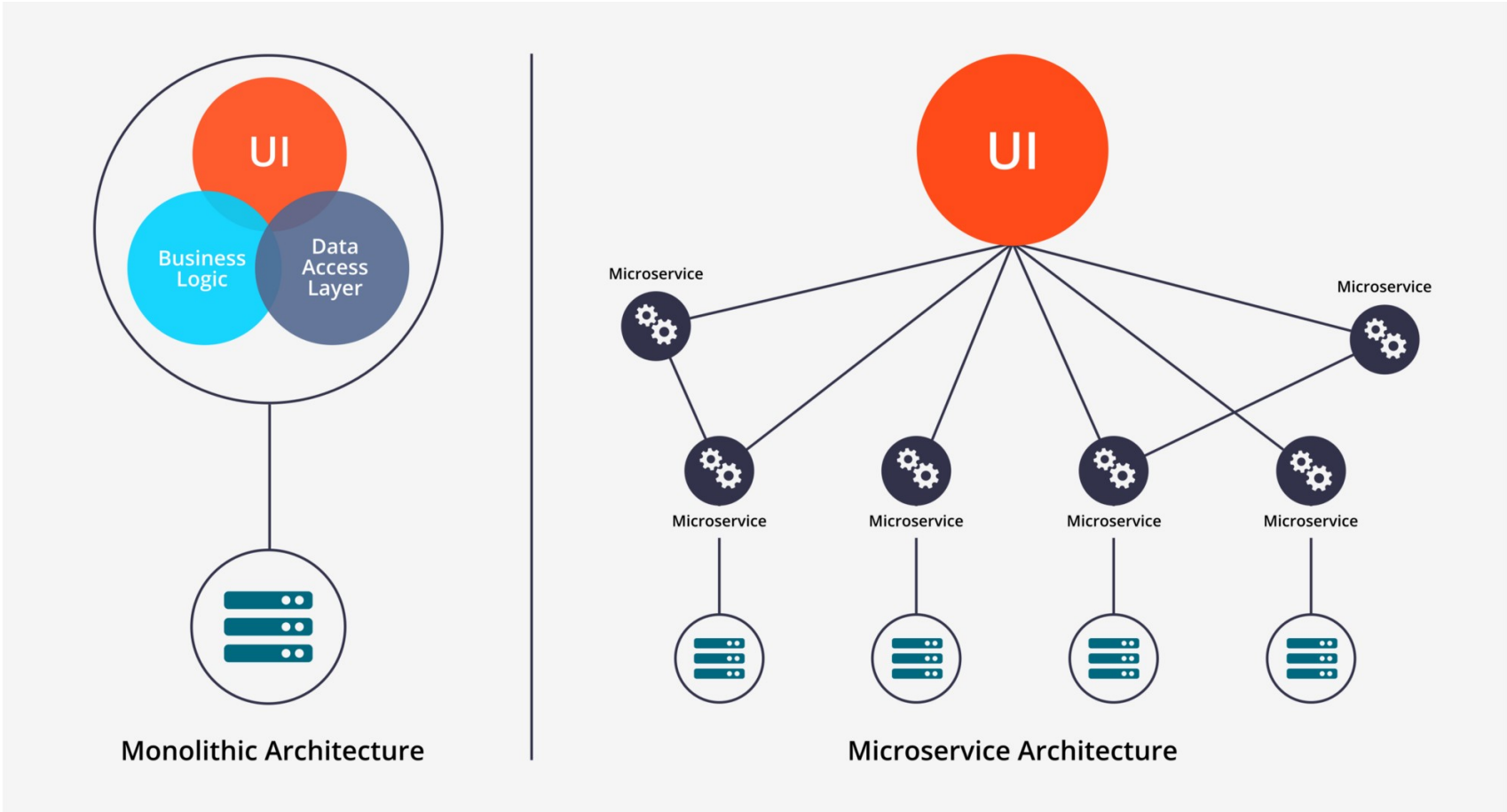
Waterfall



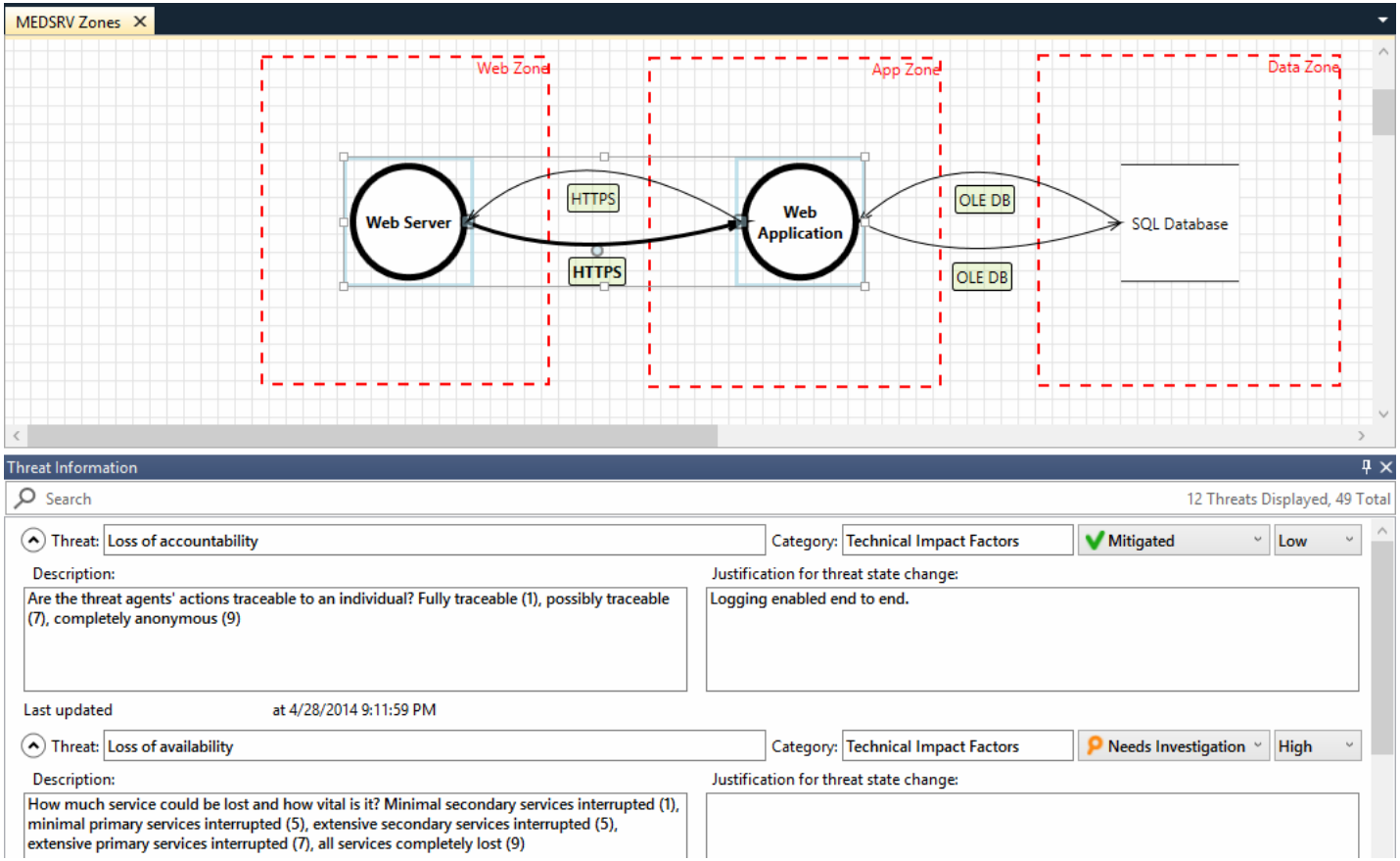
Agile



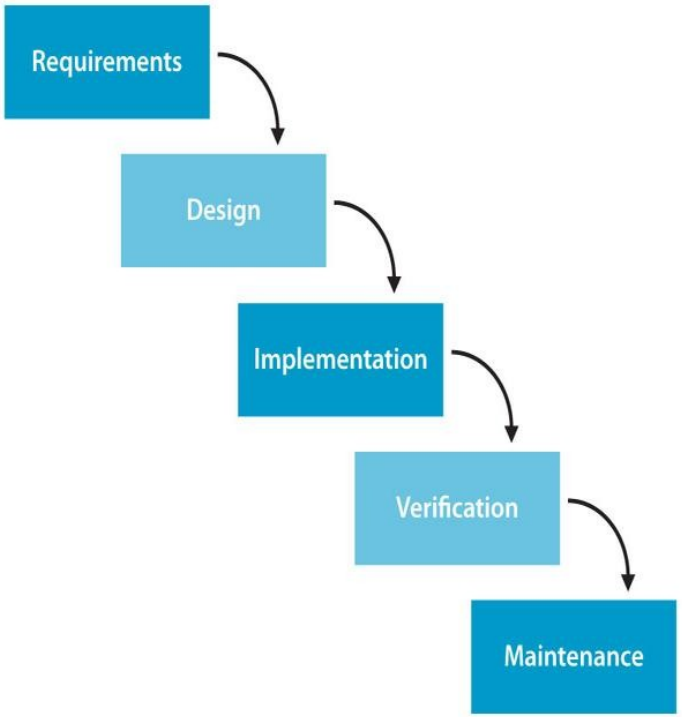
From monoliths to services

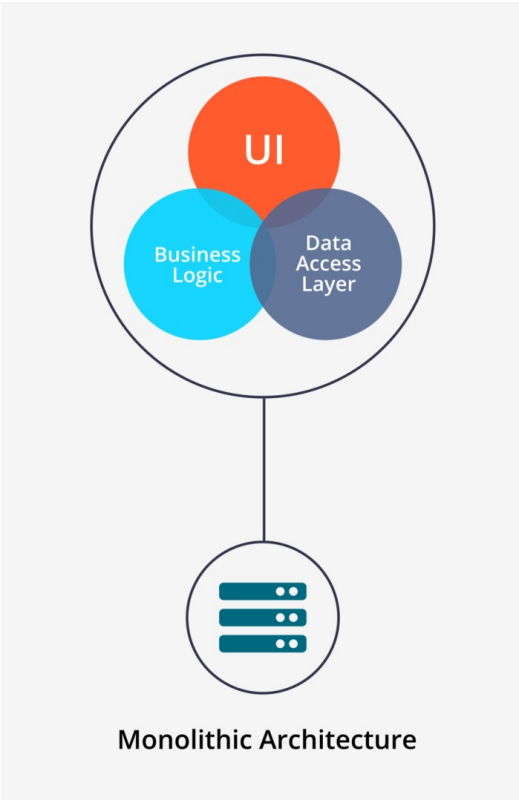


Modeling threats today



Traditional TM assumptions

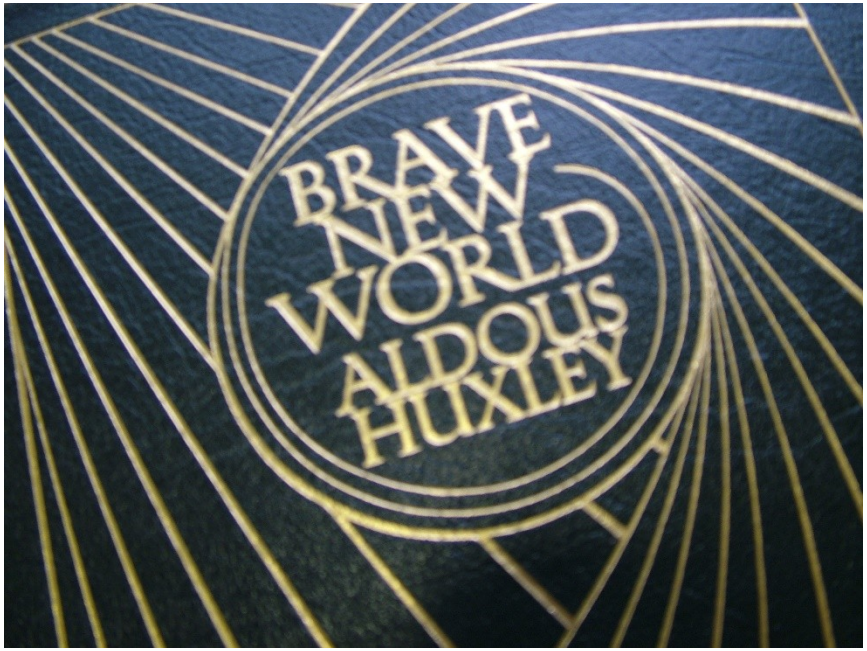




New reality

- Frequent delivery
- Working software
- New requirements
- Face to face meetings

- Independent development
- Independent deployment
- Outsourced functionality to third party services



TM becomes challenging

1. Characterize the system
   - Keep the model up to date
   - Reflect implementation details
2. Identify the threats
   - Threats can emerge, change of vanish
   - Deriving threats is slow
3. Threat and Risk analysis
   - Compositionality of services
4. Validate
   - Lack of information to automate testing

Opportunities

Agile provides grounds for

- Solid and iterative progress
- Effective analysis of complex problems

Services enable

- Verbose documentation
- Parallelization

Conclusions and open problems

- Threat Modeling can help to comply with GDPR
- Software landscape has changed, traditional TM is challenging
- TM methodologies need to take advantage of the new opportunities

- Can we automate privacy threat modeling
- Can we do *Privacy as a service*?