Digital contracts are protocols that describe and enforce execution of a contract, often among mutually distrusting parties. Programming digital contracts comes with its unique challenges, which include describing and enforcing protocols of interaction, analyzing resource usage, tracking linear assets and accounting for fault tolerance. This thesis designs programming languages and techniques for implementation and analysis of digital contracts. The core of the language is based on resource-aware session types. The talk first formalizes resource-aware session types and describes their application to sequential complexity analysis of concurrent programs. Second, the talk describes how session types can be applied to the design of Nomos, a domain-specific language for implementation of digital contracts. Nomos uses shared session types to ensure clients interact with contracts in mutual exclusion, tracks assets using a linear type system, and derives bounds on resource usage using resource-aware types. As proposed work, I will explore the practical challenges posed by session types, and the usability of Nomos. My goal would be to design an implementation of Nomos and demonstrate its efficacy in programming real world digital contracts. Furthermore, I will present my work on temporal session types for time analysis of concurrent programs, and examine if they can be used to implement timeouts in Nomos contracts.

Thesis Committee:
Jan Hoffmann, Chair
Frank Pfenning
Bryan Parno
Andrew Miller, UIUC
Shaz Qadeer, Facebook