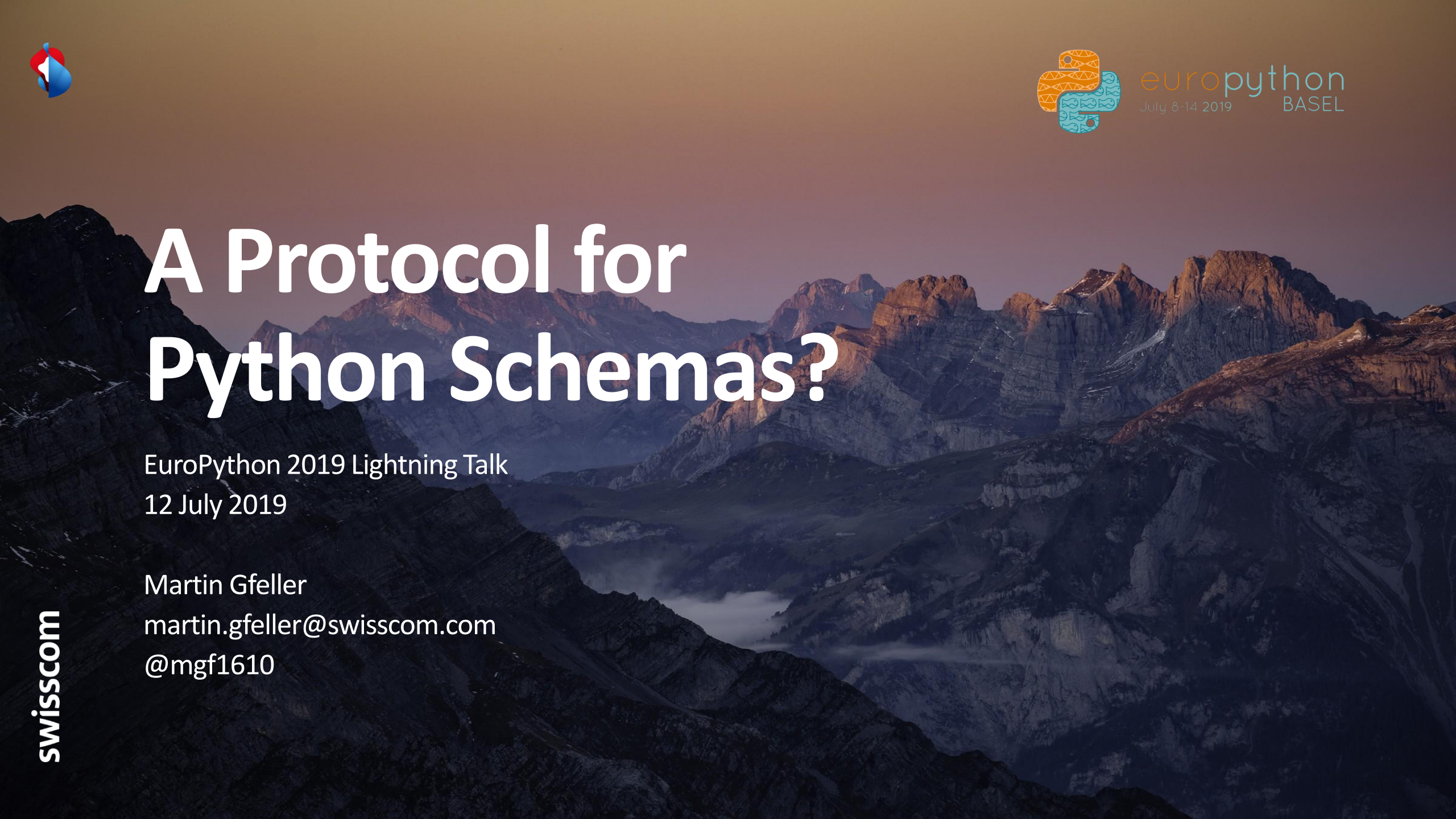


A Protocol for Python Schemas?

EuroPython 2019 Lightning Talk
12 July 2019

Martin Gfeller
martin.gfeller@swisscom.com
@mgf1610

swisscom

Schemas

Schema: Description of data objects crossing program boundaries for validation (rejecting unforeseen data) and conversion to and from other formats.

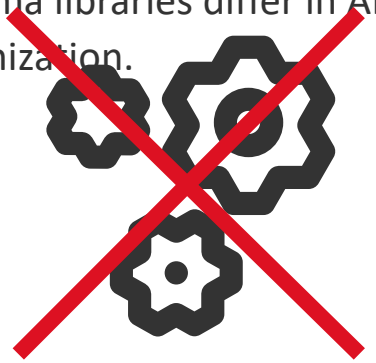
Data Types / Type Hinting is for programs – Schemas are for data exchange.

Schema Library: A library supporting definition and application of Schemas, especially conversions.

Integrated with Mapper	Standalone
Django	Cerberus
SQLAlchemy	Colander
Graphene for GraphQL	Flatland
Strawberry for GraphQL	KIM
	Marshmallow
	Schema
	Valideer
	Voluptuous

Lack of standards and interoperability:

Schema libraries differ in APIs and internal organization.



But all rely on some data types.

A Protocol for Schema Libraries

No standard API for creating Schemas or their elements, nor for internal structure.

But a minimal common protocol for interoperability:

- Schema and Data:
  - Associate a Schema to an object
  - Validate data to conform to the schema
  - Convert data to/from an external representation
  - Identify some standard external representations
- Schema handling
  - Enumerate elements of a Schema
  - Convert a Schema Element to a Python type, using PEP 593 Annotated types for additional information
  - Create a Schema Elements from a PEP 593 Annotated type created by this protocol
    - Provides a minimal “round trip” between different Schema Libraries

Interested? Get in touch!

Work in Progress: https://github.com/mcgfeller/py-schemas