# Can Persistence Become a Primary Programming Issue?

Luc Bläser, Swiss Federal Institute of Technology, Zurich

ETH Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

## Vision of Persistent Programming

## Open Challenges

No Persistence-specific artifacts

Language with Seamless Persistence Integration

Runtime System with Efficient Persistence Support

Interoperability with transient programs

Complete cache-aware garbage collection

# Persistent Active Oberon

## Programming Language with Intrinsic Persistence

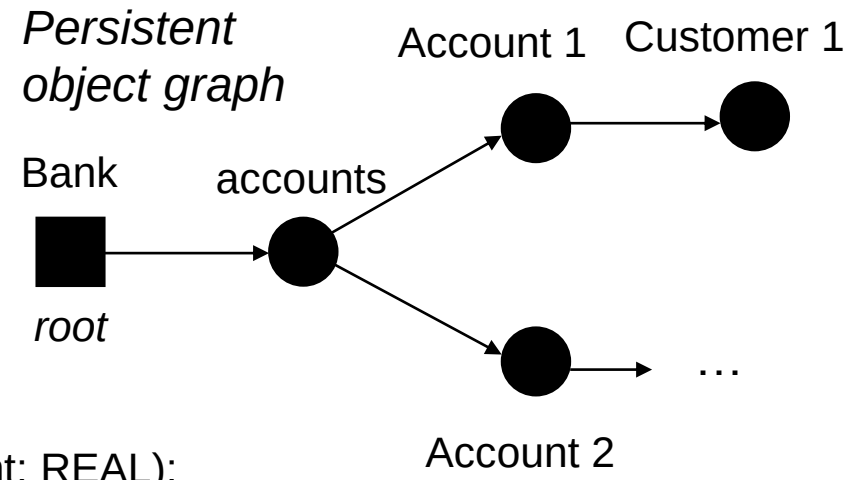
Module as persistence root

```
MODULE Bank;
  Account = OBJECT
    VAR customer: Customer; balance: REAL;
  END Account;

  VAR accounts: AccountList;

  PROCEDURE Transfer(from, to: Account; amount: REAL);
  BEGIN from := from - amount; to := to + amount
  END Transfer;

  AccountManager = OBJECT
  BEGIN {ACTIVE}
    WHILE (* … *) DO
      Transfer(a, b, c)
    END
  END AccountManager;
END Bank.
```

Persistent reference

Procedure as atomic transaction

Activity implements transaction sequence




Persistent program code ≡ transient program

# Persistent Active Oberon

Operating System with Inbuilt Persistence Support

**Memory Safety** with complete incremental garbage collection and simultaneous caching

**Interoperability** between persistent and transient modules (via implicit transient references)

**Performance and scalability** is well competitive to classical implementations (using databases)

Project webpage for system download:
*http://www.bluebottle.ethz.ch/Persistent*