Lightning Talk
Tree transformation with Python

Robert Niederreiter <office@squarewave.at>

BlueDynamics Alliance

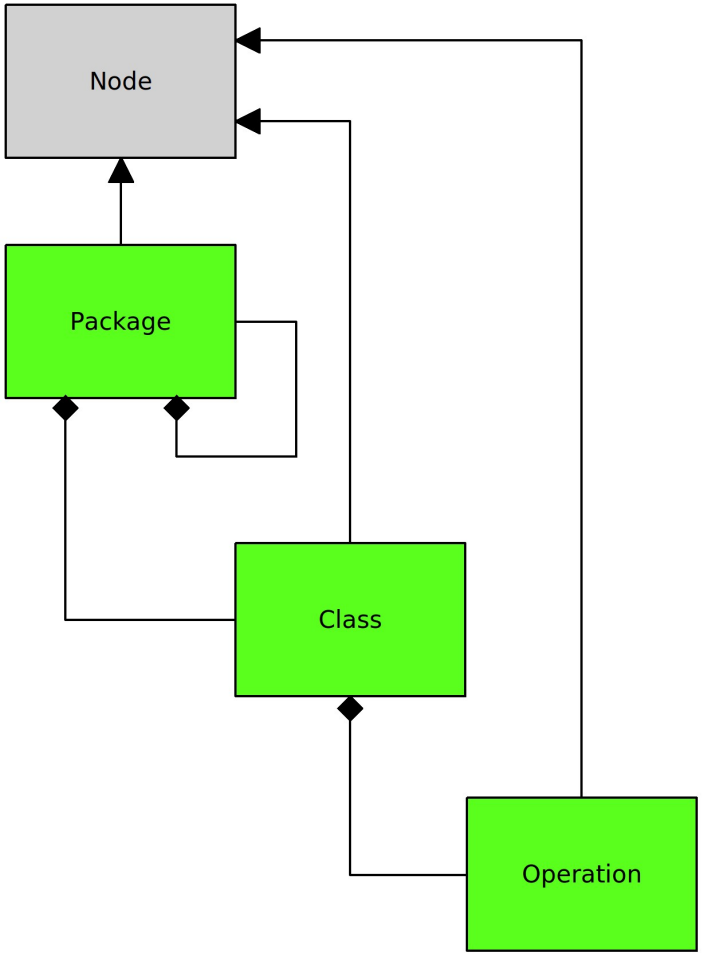
Nodes and trees

- Node
  - Base object for building trees
  - Provides a dictionary like API and implements zope.interface.mapping.IFullMapping
  - Knows about it's position. It implements zope.location.interfaces.ILocation
  - Can be accessed flat. Nodes are accessible from each other via UID references
- Every data structure can be represented as tree, i.e. XML, UML, File System, SQL, LDAP

Robert Niederreiter <office@squarewave.at>

BlueDynamics Alliance

UML as node compliant tree

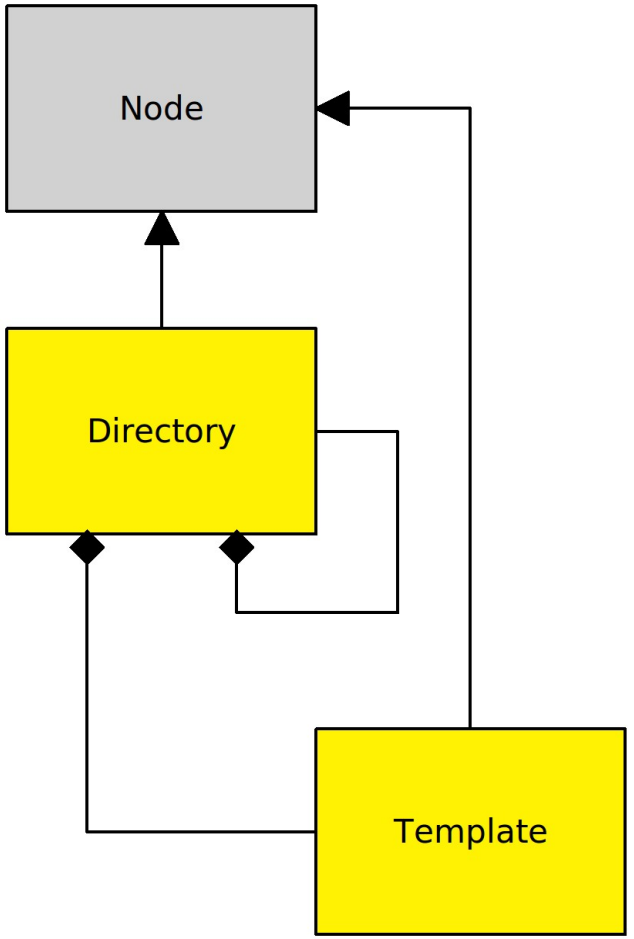


* Many more...

Robert Niederreiter <office@squarewave.at>

BlueDynamics Alliance

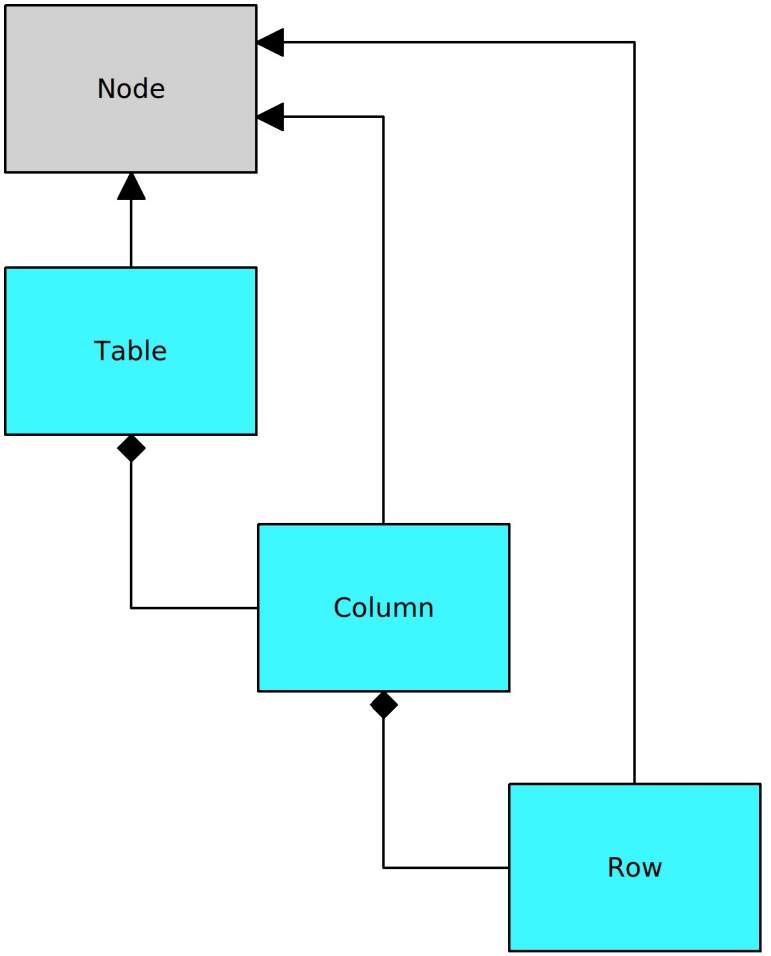
FS as node compliant tree



Robert Niederreiter <office@squarewave.at>

BlueDynamics Alliance

SQL as node compliant tree



Robert Niederreiter <office@squarewave.at>

BlueDynamics Alliance

IO System

- AGX IO System
  - Based on Nodes
  - Follows the interfaces provided by Node
  - Introduces ISource and ITarget, both derive from INode
  - ITarget promises to persist or dump the tree on __call__() in any way.

Robert Niederreiter <office@squarewave.at>

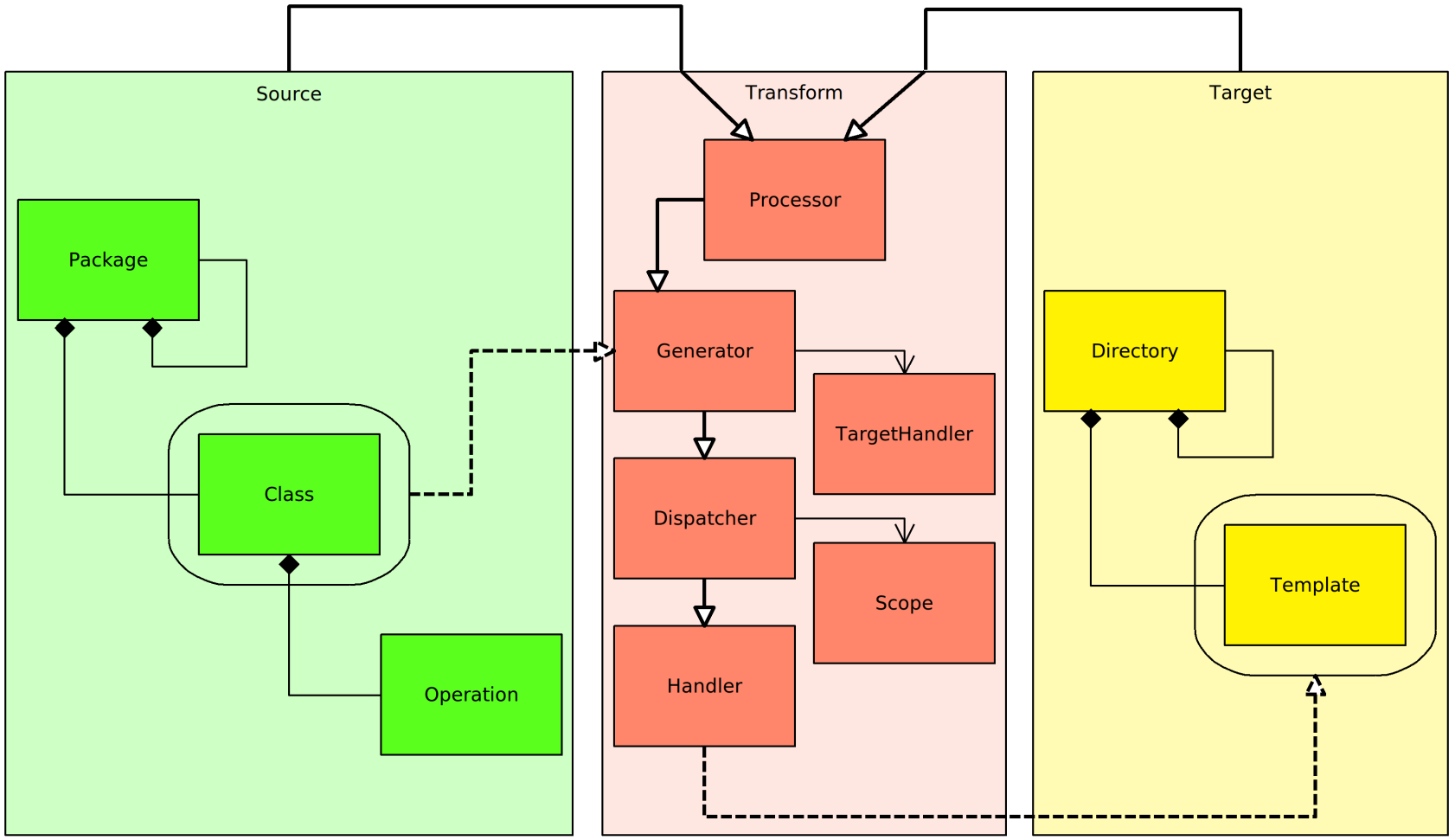
BlueDynamics Alliance

Tree transformation chain

- Iterate over all nodes of source tree and look up handlers, which are responsible for creating target nodes inside the target tree based on the information from the source node.
- Transformation chain components located in agx.core
- Finally target tree is called.

Robert Niederreiter <office@squarewave.at>

Tree transformation chain



Robert Niederreiter <office@squarewave.at>

BlueDynamics Alliance

Thank you!

Robert Niederreiter <office@squarewave.at>

BlueDynamics Alliance