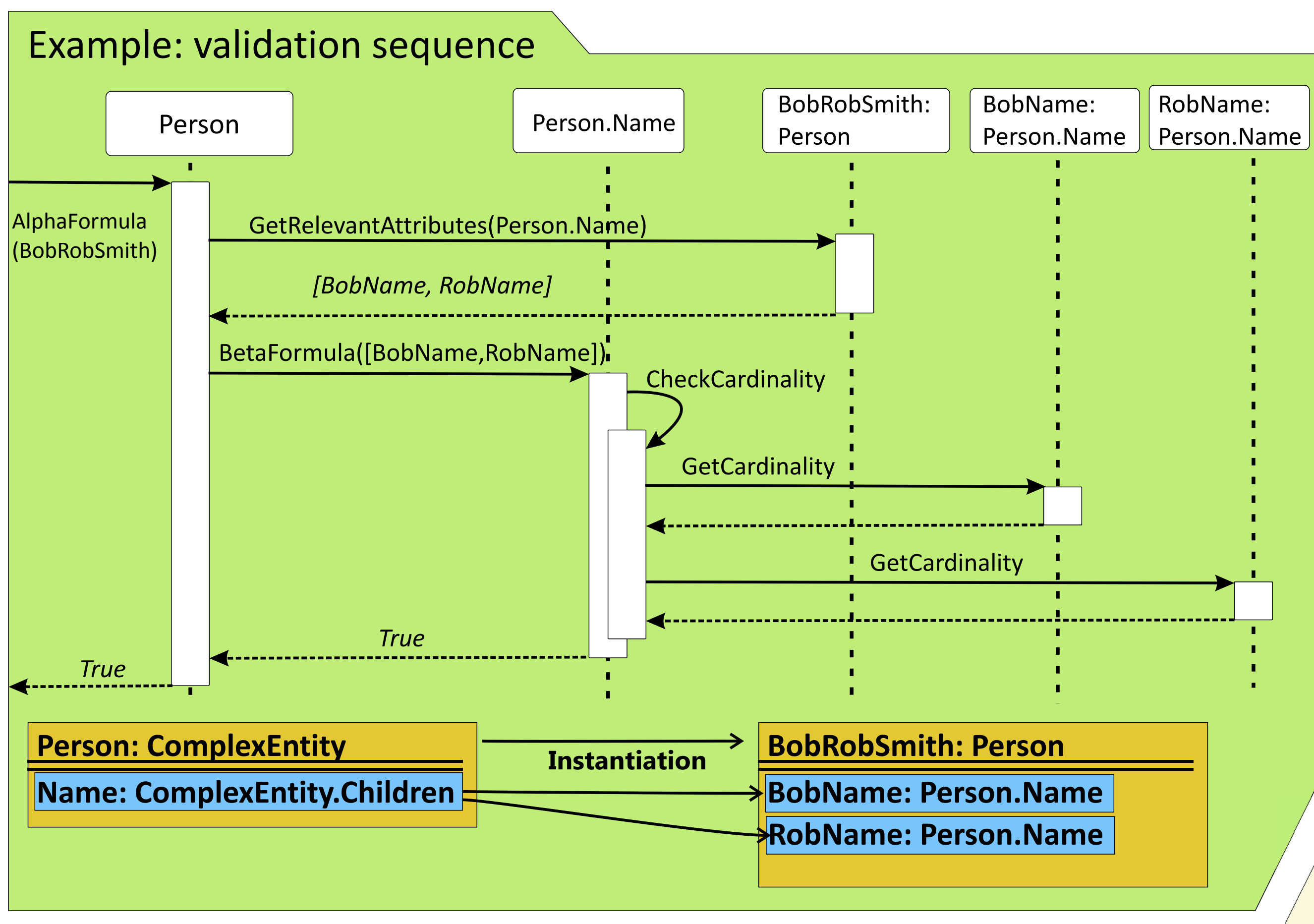
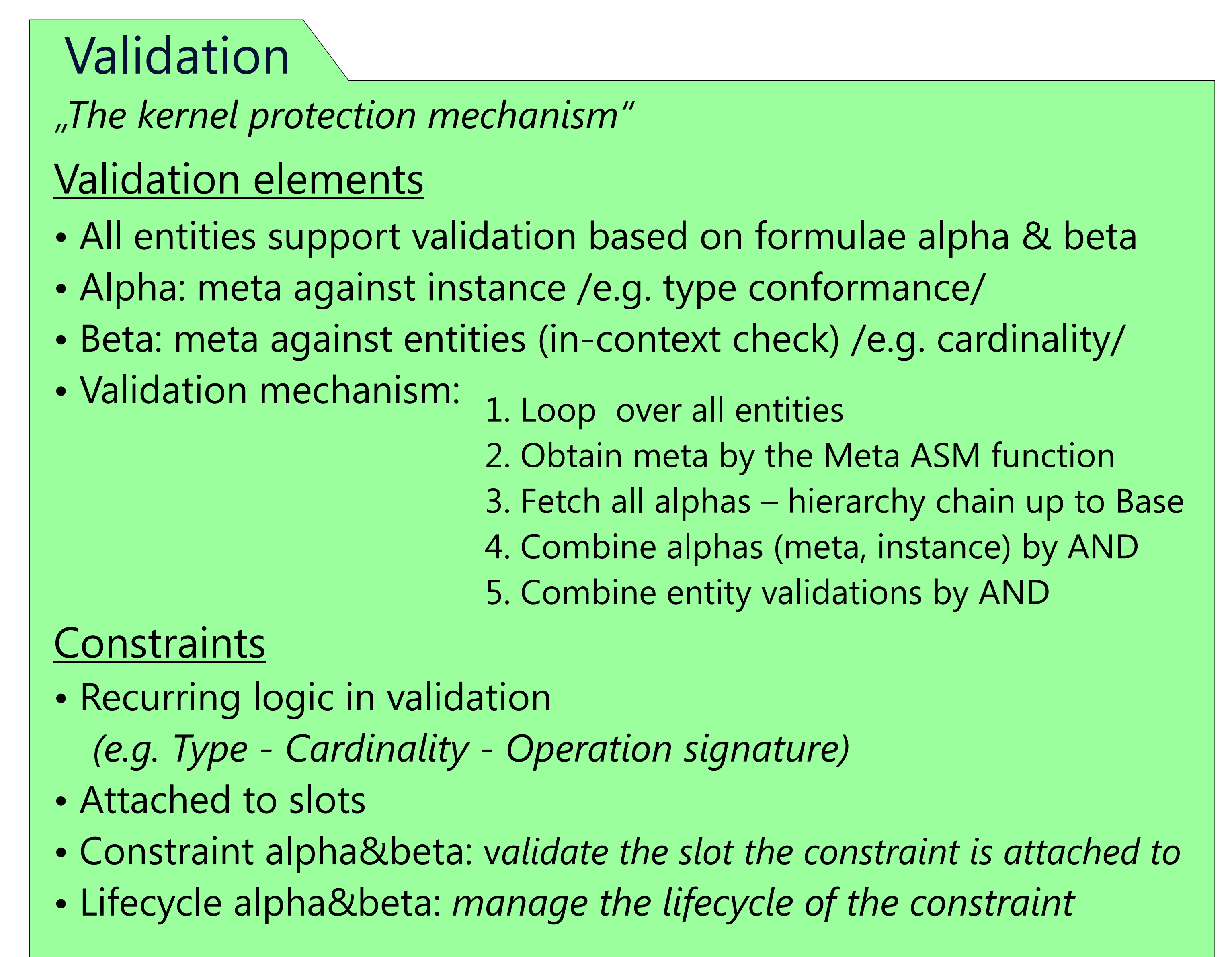
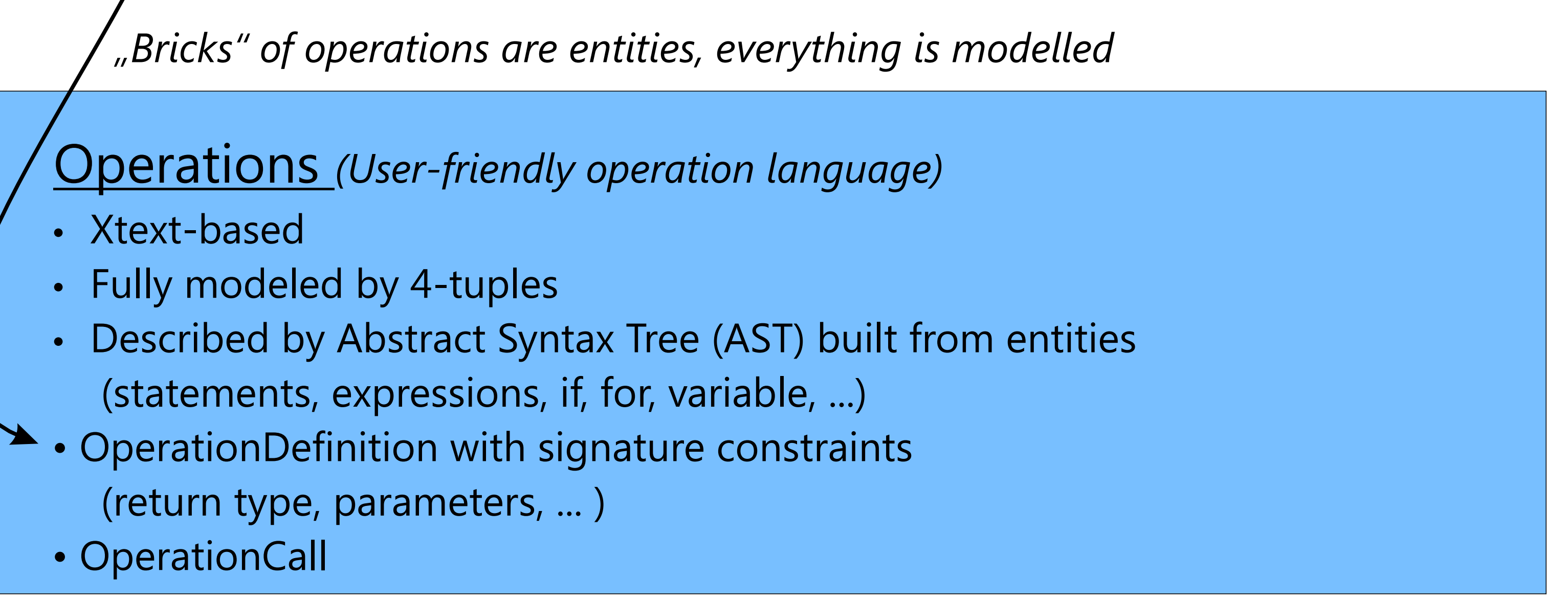
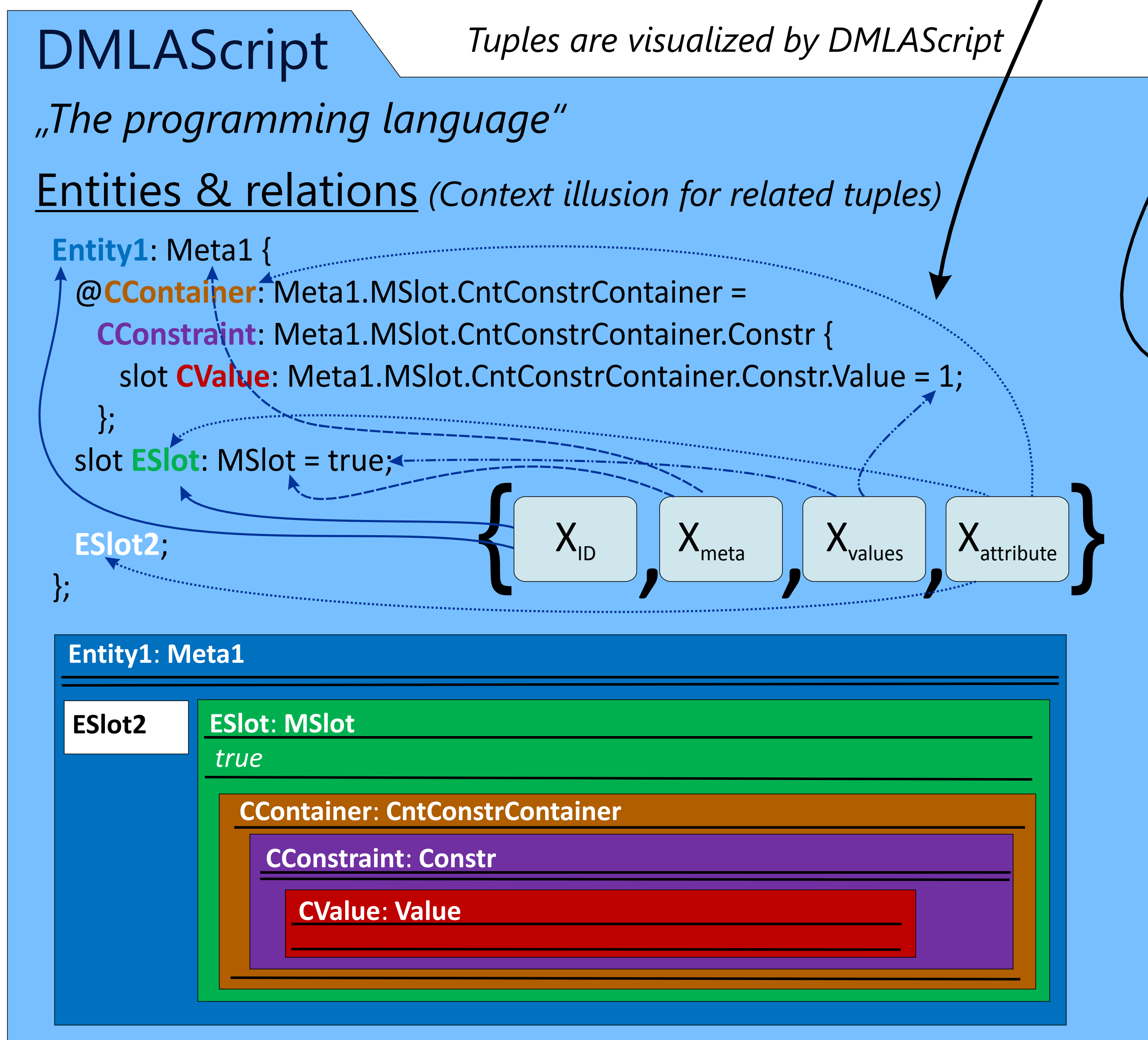
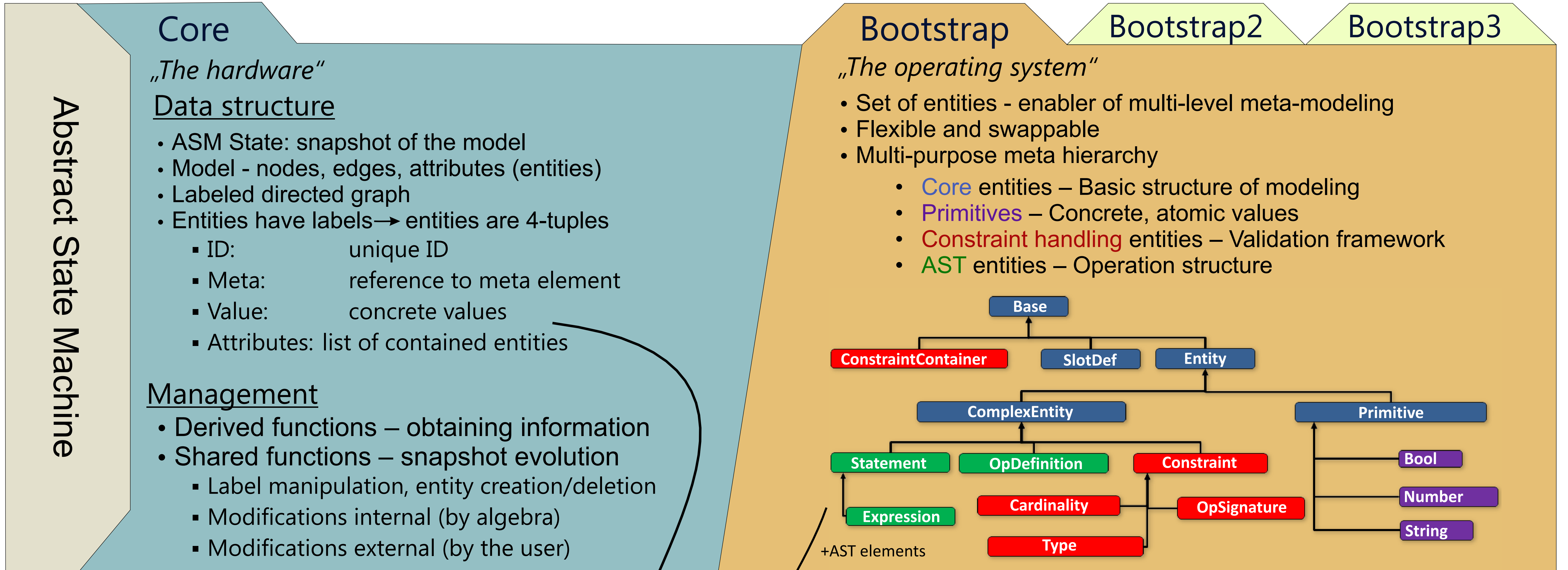


Self-Describing Operations for Multi-level Meta-modeling



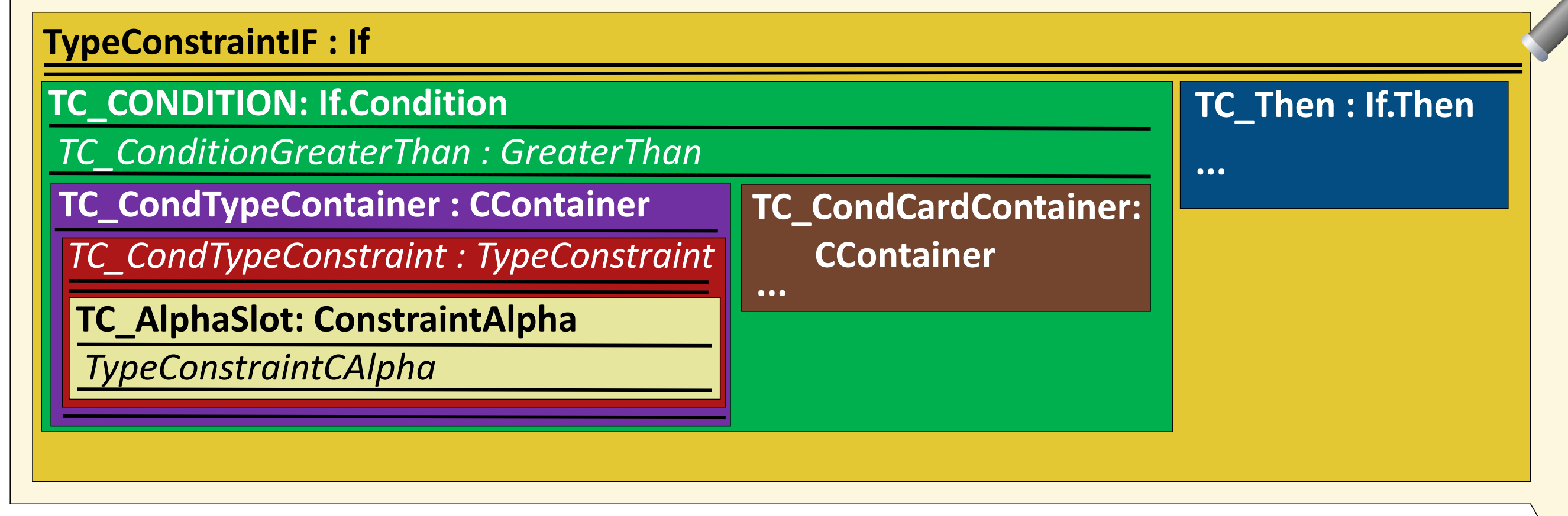
Dániel Urbán (DAAI, Budapest University of Technology and Economics, Budapest, Hungary)
 Zoltán Theisz (evopro systems engineering Ltd., Hauszmann Alajos str. 2, Budapest, Hungary)
 Gergely Mezei (DAAI, Budapest University of Technology and Economics, Budapest, Hungary)



```

Example: validation code
operation Bool ID::TypeConstraintAlpha(ID meta, ID instance) {
    ID type = call this::Helper.GetTypeType();
    Bool isInclusive = call this::Helper.GetTypeIsInclusive();
    Bool unfilled = type==null || isInclusive==null;
    ID[] typeConstraints = call instance::Helper.GetRelevantConstraints($Bootstrap.Type);
    if (size(typeConstraints)>1) return false;
    Object[] values = call $Values(instance);
    if (values==null || size(values)==0) return true;
    if (unfilled && size(typeConstraints)==0) return false;

    ID actualType = type;
    Bool actualIsInclusive = isInclusive;
    if (size(typeConstraints)==1) {
        ID instTypeConstraint = index<ID>(typeConstraints, 0);
        actualType = call instTypeConstraint::Helper.GetTypeType();
        actualIsInclusive = call instTypeConstraint::Helper.GetTypeIsInclusive();
    }
    if (actualType==null || actualIsInclusive==null) return false;
    for (Object value : values) {
        if (actualIsInclusive) {
            if (!call $DerivesFromOrEquals(actualType, value)) return false;
        } else {
            if (!call $DerivesFrom(actualType, value)) return false;
        }
    }
    return true;
}
    
```



<http://www.aut.bme.hu/Pages/Research/VMTS/DMLA>