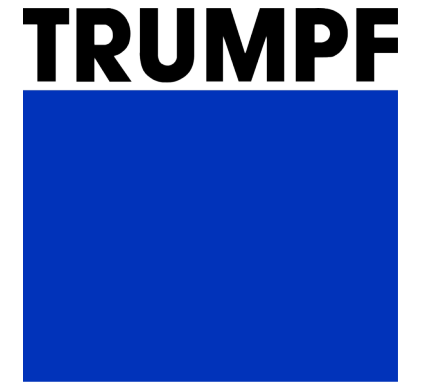


Real-world LionWeb at TRUMPF



Niko Stotz, F1RE

Johannes Tauber, TRUMPF



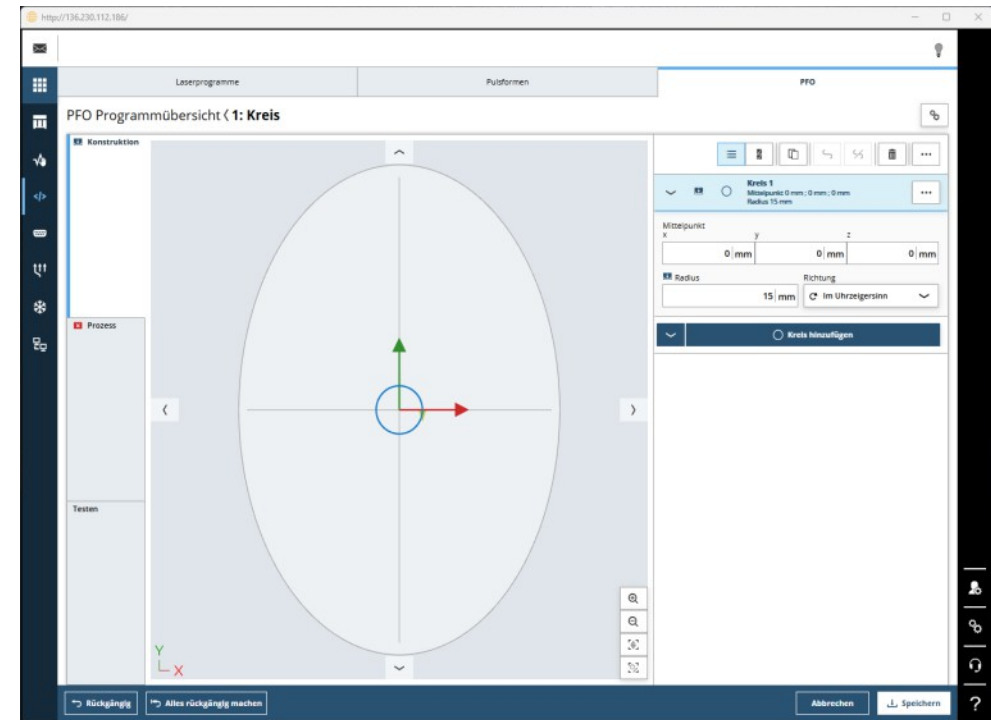
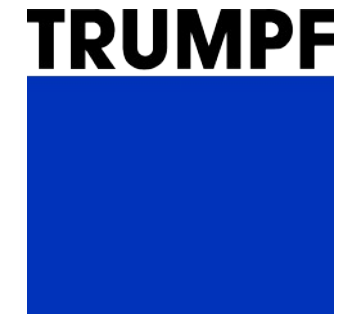
Takeaways

- DSLs work in real life
- LionWeb fulfills real-world needs

Why did
TRUMPF start
this?



Outdated software



Challenge: Ambiguous or missing terms



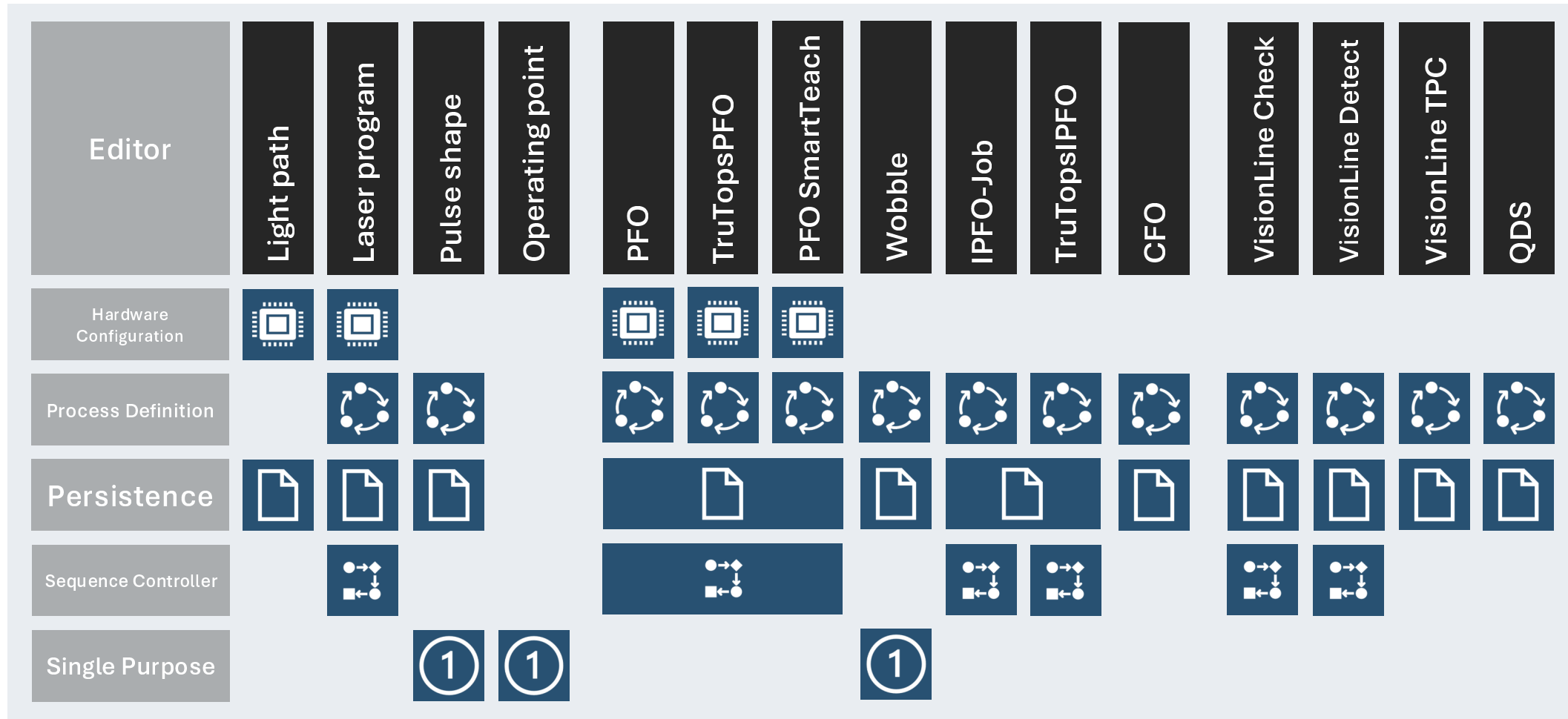
- Ambiguous:
 - Path vs. Geometry vs. Shape vs. Contour vs. Hot* / Cold*
 - Wobble program vs. wobble template vs. wobble type vs. wobble kind
 - 2 x process parameter
- Missing:
 - Processing step
 - Arguments (as in program call)
 - Process value

Solution: Ubiquitous language



- Disambiguation
 - High-level geometry: As seen by the user
 - Primitive geometry: Can not be deconstructed further in a mathematically sense
 - Path: Can be processed by laser hardware
- Guard uniform usage
 - UI / docs / code
 - Pervasive in discussions
 - Easy to connect to unknown details

Challenge: Different configs, definitions, persistence per editor



Solution: DSL + LionWeb serialization



```
> [L] TL.LDM.Language.Definition.Application
> [L] TL.LDM.Language.Definition.Application.Analysis
> [L] TL.LDM.Language.Definition.Base
> [L] TL.LDM.Language.Definition.Capability
> [L] TL.LDM.Language.Definition.DataSource
> [L] TL.LDM.Language.Definition.Finding
> [L] TL.LDM.Language.Definition.Geometry.Base
> [L] TL.LDM.Language.Definition.Geometry.Construction
▼ [L] TL.LDM.Language.Definition.Geometry.HighLevel
  ▼ [S] structure
    ▼ [S] geometry
      > [S] AbstractHighLevelGeometry
      > [S] Arc
      > [S] Circle
      > [S] Ellipse
      > [S] EllipseArc
```

```
{
  "serializationFormatVersion": "2023.1",
  "languages": [
    {
      "key": "LionCore-M3",
      "version": "2023.1"
    },
    {
      "key": "LionCore-builtins",
      "version": "2023.1"
    },
    {
      "key": "io-lionweb-mps-specific",
      "version": "0"
    }
  ],
  "nodes": [
    {
      "id": "TL-LDM-Base",
      "classifier": {
        "language": "LionCore-M3",
        "version": "2023.1",
        "key": "Language"
      },
    },
  ],
}
```

Just implement it?



Challenge: No tooling for prototyping



- Browser-based UI not yet available
- API usage tedious

```
var refinement = new HighLevelConstructionRefinement("id1")
{
    Name = "myRefinement",
    Geometries =
    [
        new Line("id2")
        {
            Start = new CartesianCoordinate("id3")
            {
                X = new LengthValue("id4") { Value = new NumberLiteral("id5") {
                    Value = "1" }, Exponent = 0 },
                Y = new LengthValue("id6") { Value = new NumberLiteral("id7") {
                    Value = "2" }, Exponent = 0 },
                Z = new LengthValue("id8") { Value = new NumberLiteral("id9") {
                    Value = "0" }, Exponent = 0 }
            },
            End = new CartesianCoordinate("id10") { }
        }
    ]
};
```

Solution: MPS + Spikes



high-level refinement myRefinement

arguments: << ... >>

geometries: line myLine [1 · 10⁰ m, 2 · 10⁰ m, 0 · 10⁰ m] -> [0 · 10⁰ m, 3 · 10⁰ m, 0 · 10⁰ m]

- Also very helpful to show vision

[construction = construction<g>()|processingUnit = ProcessingUnit01] wobble <none> sensoric <none>

pfo program main()

with program settings {

power request: absolute power mode

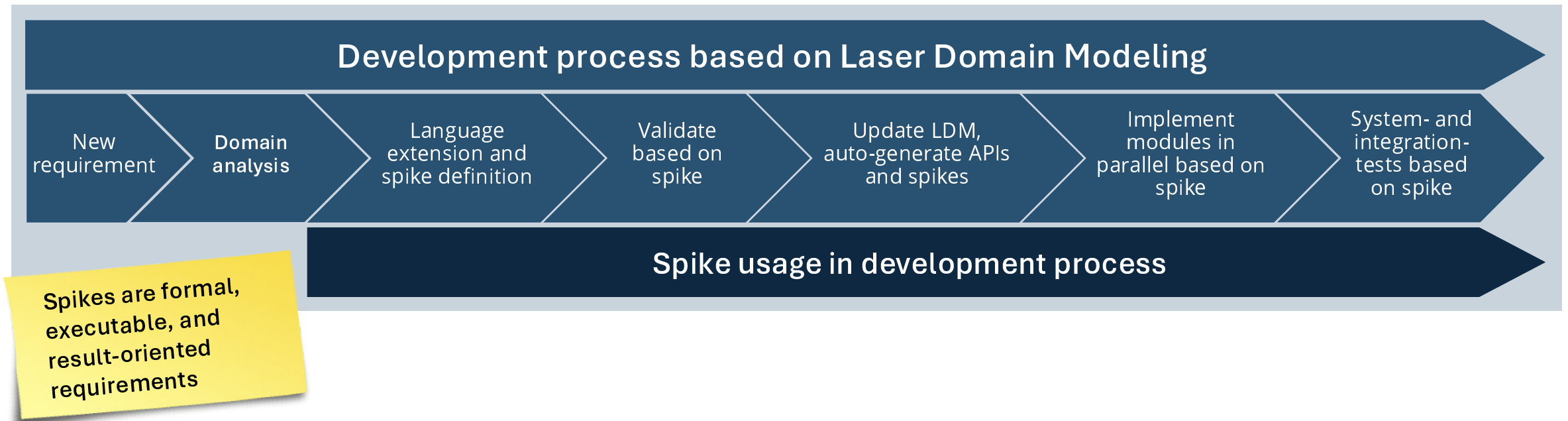
}

{

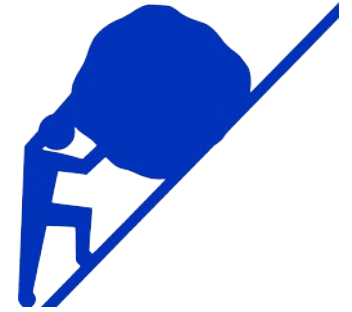
step extend [offset [iterate [reverse [construction.path_rectangle_1] : 3 · 10⁰ : 4 · 10⁻³ m] : 2 · 10⁻³ m]

Standardized feature development process

- Minimize risks through iterative development based on spikes



Challenge: Support C# for backend, TypeScript for frontend



- Languages established at TRUMPF
- Needs to integrate with other software on laser
 - functionally
 - visually

Solution: Export structure from MPS, generate to target language



```
/**
 * A cartesian coordinate, expressed with x, y and z
 * A cartesian coordinate, expressed with x, y and z as length values
 */
export class CartesianCoordinate extends AbstractCoordinate implements LdmBase.IInterprete
static create(id: $lwJson.LionWebId, receiveDelta?: $lwClassCore.DeltaReceiver, pa
    $lwClassCore.Parentage): CartesianCoordinate {
    return new CartesianCoordinate(LdmGeometryBaseBase.INSTANCE.CartesianCoordinate,
}
private readonly _x: $lwClassCore.RequiredSingleContainmentValueManager<LdmBase.IL
/**
 * The x length value of the coordinate
 */
get x(): LdmBase.ILengthValue {
    return this._x.get();
}
set x(newValue: LdmBase.ILengthValue) {
    this._x.set(newValue);
}
```



```
@doc A cartesian coordinate, expressed with x, y and z as length values
concept CartesianCoordinate extends AbstractCoordinate
    implements IInterpreterResult

instance can be root: false
alias: [
    short description: A cartesian coordinate, expressed with x, y and z

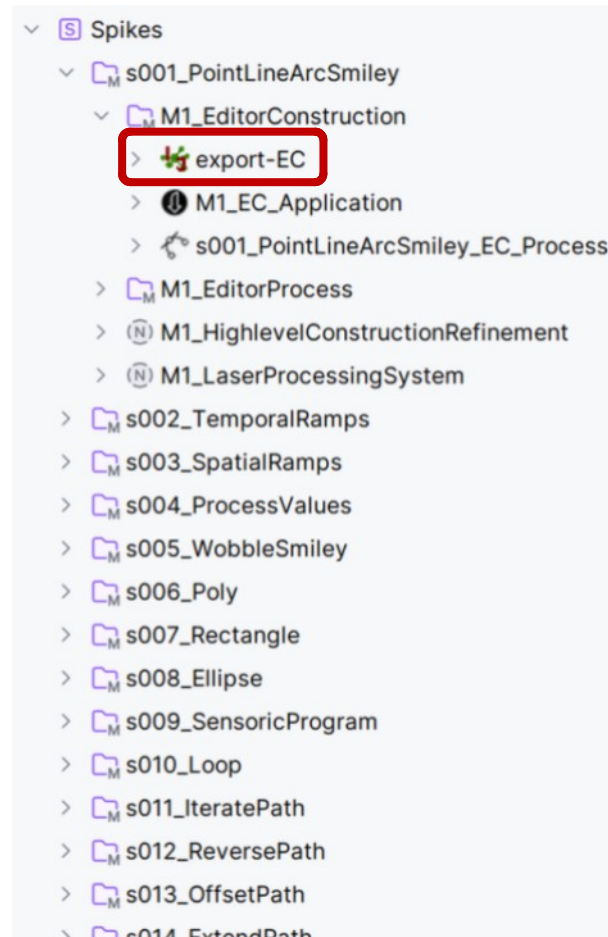
properties:
    << ... >>

children:
    @doc The x length value of the coordinate
    x : ILengthValue[1]
        LionWeb key: CartesianCoordinate-x
    @doc The y length value of the coordinate
    y : ILengthValue[1]
        LionWeb key: CartesianCoordinate-y
    @doc The z length value of the coordinate
    z : ILengthValue[1]
        LionWeb key: CartesianCoordinate-z
```

```
/// <summary>A cartesian coordinate, expressed with x, y and z</summary>
/// <summary>A cartesian coordinate, expressed with x, y and z as length
[LionCoreMetaPointer(Language = typeof(BaseLanguage), key = "CartesianCoo
[Alias("[")]]
public sealed partial class CartesianCoordinate : AbstractCoordinate,
    TL.LDM.Language.Definition.Base.IInterpreterResult
{
    private TL.LDM.Language.Definition.Base.ILengthValue? _x = null;
    /// <summary>The x length value of the coordinate</summary>
    /// <remarks>Required Single Containment</remarks>
    /// <exception cref = "UnsetFeatureException">If X has not been set</exception>
    /// <exception cref = "InvalidValueException">If set to null</exception>
    [LionCoreMetaPointer(Language = typeof(BaseLanguage), Key = "CartesianCoordinate-x")]
    [LionCoreFeature(Kind = LionCoreFeatureKind.Containment, Optional = false, Multiple = false)]
    public TL.LDM.Language.Definition.Base.ILengthValue X { get => _x ?? throw new
        UnsetFeatureException(BaseLanguage.Instance.CartesianCoordinate_x); set => SetX(value); }
```



Challenge: Use spikes for testing



Solution: Export models from MPS



```
{
  "serializationFormatVersion": "2023.1",
  "languages": [
    {
      "key": "TL-LDM-Geometry-Construction",
      "version": "14.0.0"
    },
    {
      "key": "TL-LDM-Base",
      "version": "14.0.0"
    },
    {
      "key": "TL-LDM-Geometry-HighLevel",
      "version": "14.0.0"
    }
  ],
  "nodes": [
    {
      "id": "3281712411898913648",
      "classifier": {
        "language": "TL-LDM-Geometry-Construction",
        "version": "14.0.0",
        "key": "HighLevelConstructionRefinement"
      },
      "properties": [
        {

```

Challenge: Debug models



```
{
  "serializationFormatVersion": "2023.1",
  "languages": [
    {
      "key": "TL-LDM-Geometry-Construction",
      "version": "14.0.0"
    },
    {
      "key": "TL-LDM-Base",
      "version": "14.0.0"
    },
    {
      "key": "TL-LDM-Geometry-HighLevel",
      "version": "14.0.0"
    }
  ],
  "nodes": [
    {
      "id": "3281712411898913648",
      "classifier": {
        "language": "TL-LDM-Geometry-Construction",
        "version": "14.0.0",
        "key": "HighLevelConstructionRefinement"
      },
      "properties": [
        {
```

Solution: Import models to MPS



Logical View

- temp_debug
 - temp_debug
 - a_model
 - Import s001

Import s001

Import Instance from JSON Import s001

`${com.trumpf.root}/LDM/mps/exports/spikes/s001_EP_PointLineArcSmiley.json`



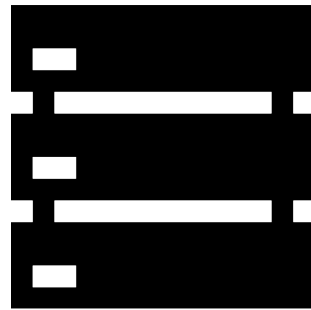
Challenge: Backend communication

frontend

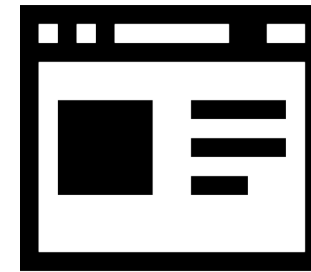
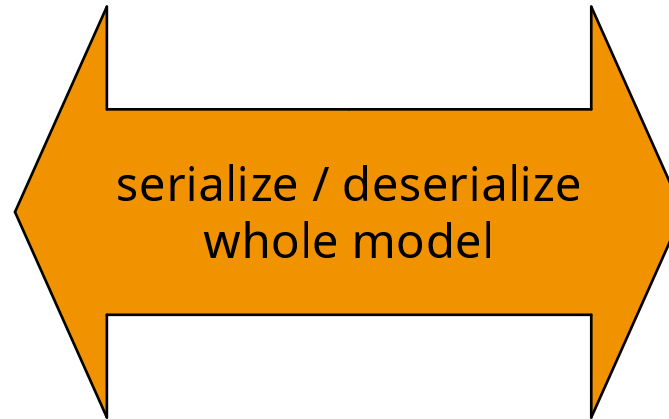


- Backend
 - Persistence
 - Discretization
- Frontend
 - User interaction
 - Rendering

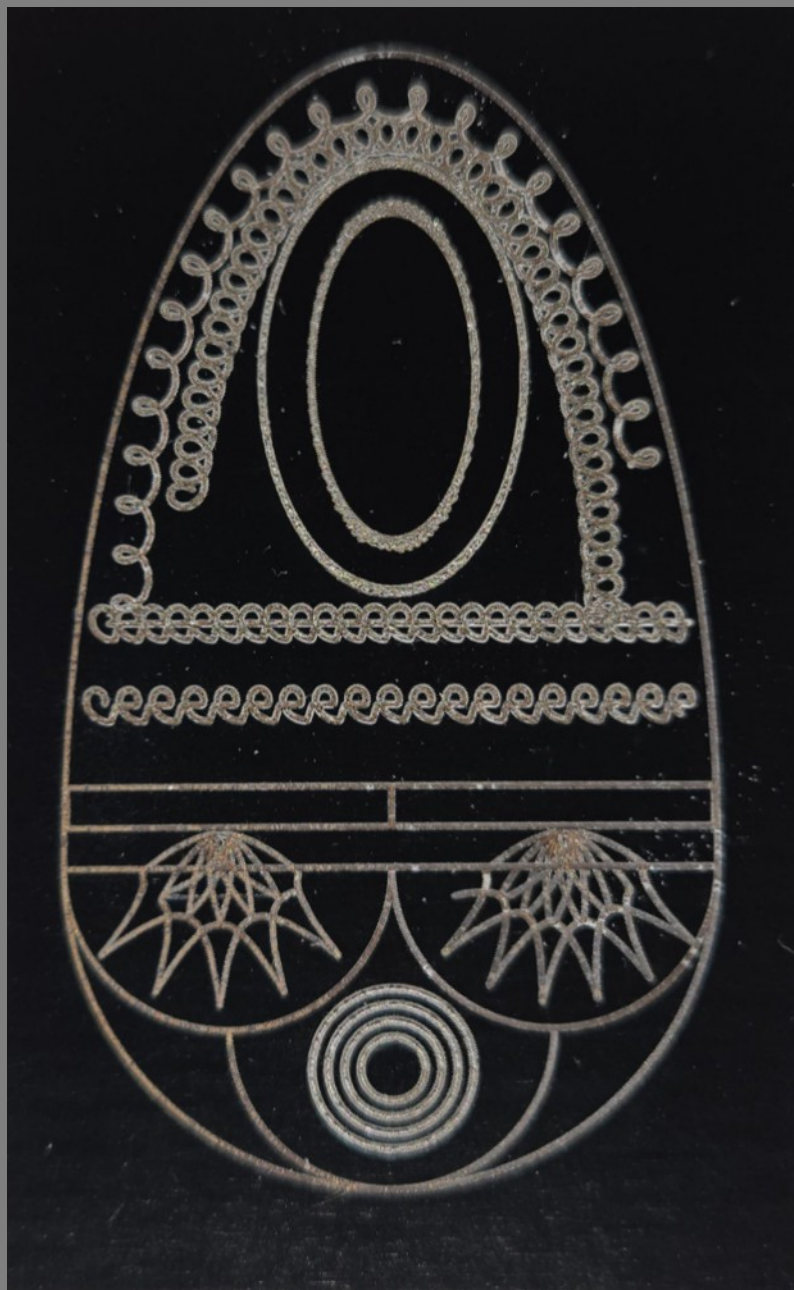
Solution: LionWeb bulk



backend



frontend



Challenge: Language evolution



```
abstract concept AbstractRealValue extends AbstractValue
    implements IRealValue

instance can be root: false
alias: <no alias>
short description: <no short description>

properties:
    exp : integer
        LionWeb key: AbstractRealValue-exp
        LionWeb optional: true

children:
    @doc The real number value
    value : Expression[1]
        LionWeb key: AbstractRealValue-value
    @Deprecated (use exp instead)
    exponent : NumberLiteral[0..1]
        LionWeb key: AbstractRealValue-exponent
```

```
public abstract partial class AbstractRealValue : AbstractValue, IRealValue
{
    private int? _exp = null;
    /// <remarks>Optional Property</remarks>
    [LionCoreMetaPointer(Language = typeof(BaseLanguage), Key = "AbstractRealValue-exp")]
    [LionCoreFeature(Kind = LionCoreFeatureKind.Property, Optional = true, Multiple = false)]
    public int? Exp { get => _exp; set => SetExp(value); }

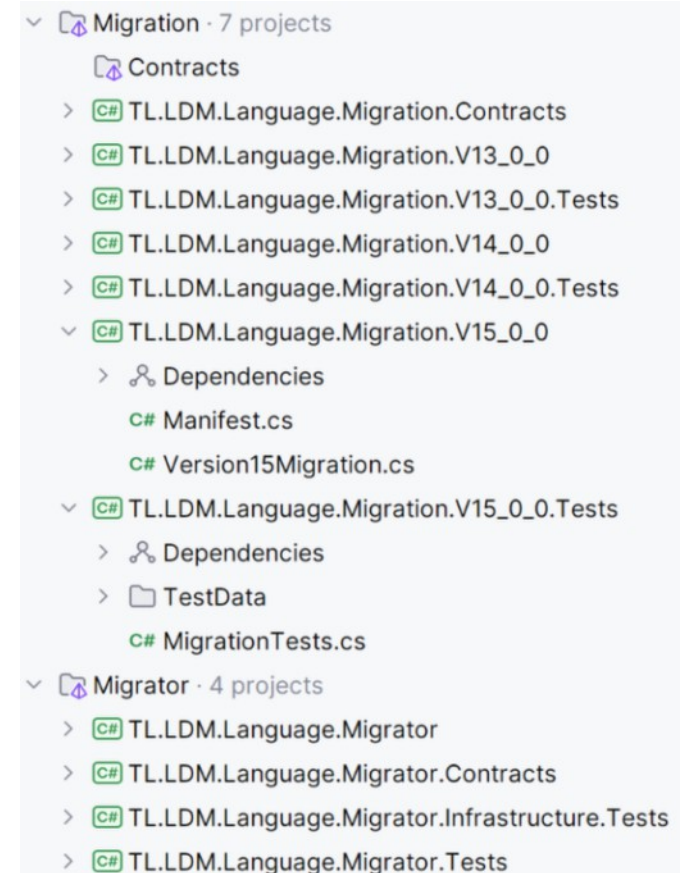
    private Org.Iets3.Core.Expr.SimpleTypes.NumberLiteral? _exponent = null;
    /// <remarks>Optional Single Containment</remarks>
    [LionCoreMetaPointer(Language = typeof(BaseLanguage), Key = "AbstractRealValue-exponent")]
    [LionCoreFeature(Kind = LionCoreFeatureKind.Containment, Optional = true, Multiple = false)]
    [Obsolete("use exp instead")]
    public Org.Iets3.Core.Expr.SimpleTypes.NumberLiteral? Exponent { get => _exponent; set => SetExponent(value); }
```

Solution: LionWeb migration



```
{
  "serializationFormatVersion": "2023.1",
  "languages": [
    {
      "key": "TL-LDM-Geometry-Construction",
      "version": "14.0.0"
    },
    {
      "key": "TL-LDM-Base",
      "version": "14.0.0"
    },
    {
      "key": "TL-LDM-Geometry-HighLevel",
      "version": "14.0.0"
    }
  ],
  "nodes": [
    {
      "id": "3281712411898913648",
      "classifier": {
        "language": "TL-LDM-Geometry-Construction",
        "version": "14.0.0",
        "key": "HighLevelConstructionRefinement"
      },
      "properties": [
        {

```



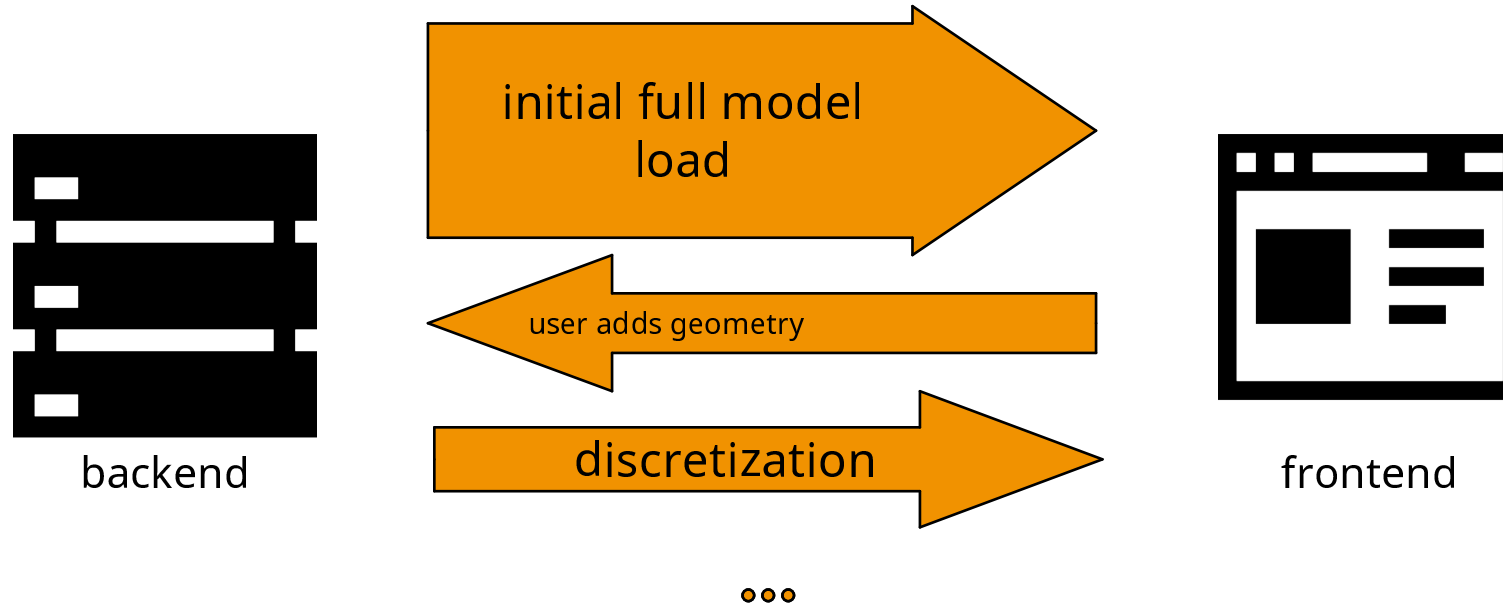


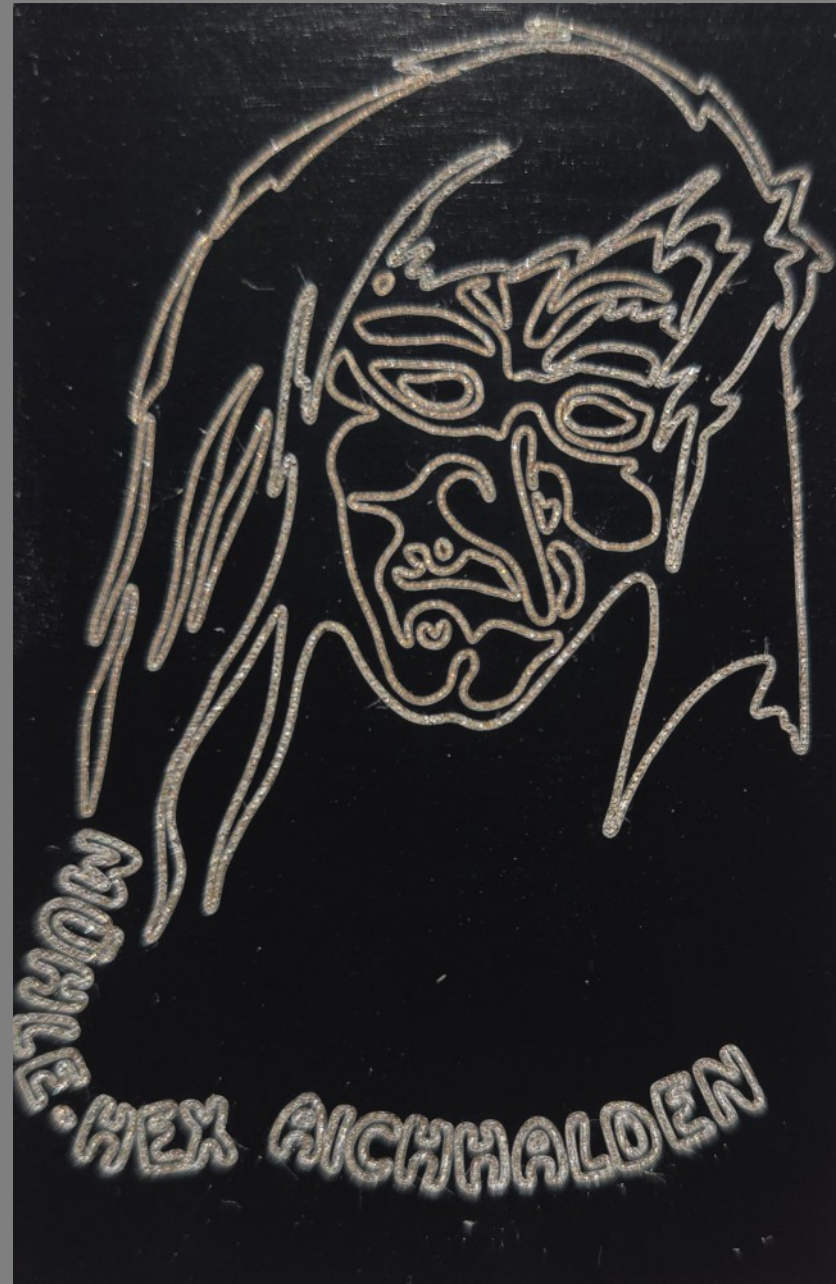
Challenge: Performance issues backend frontend sync



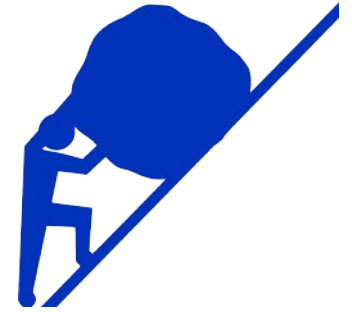
- Bigger models
 - Hundreds of geometries
- Changes on both backend and frontend

Solution: LionWeb delta



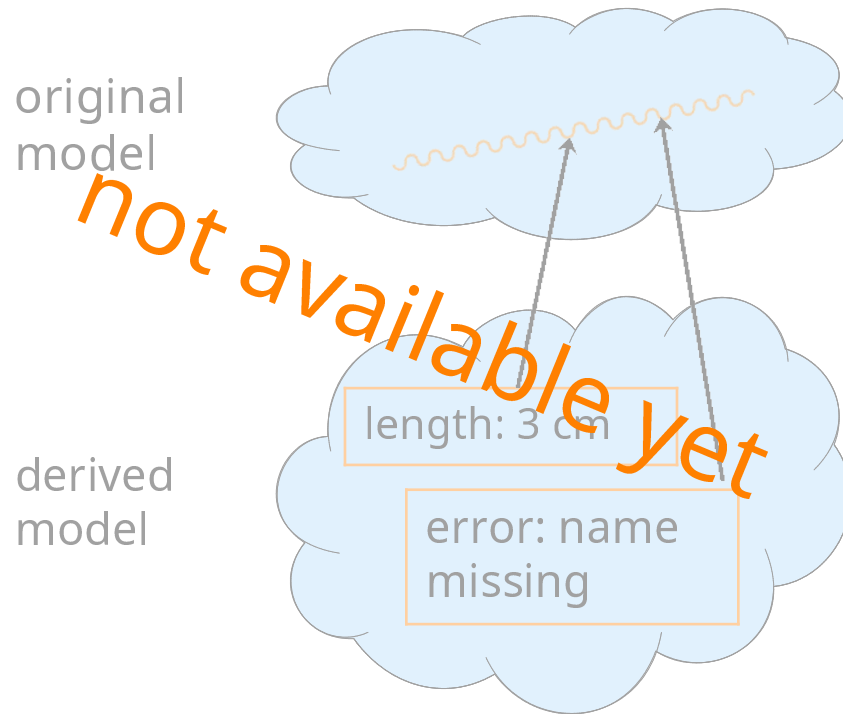


Challenge: Communicate transient information backend frontend



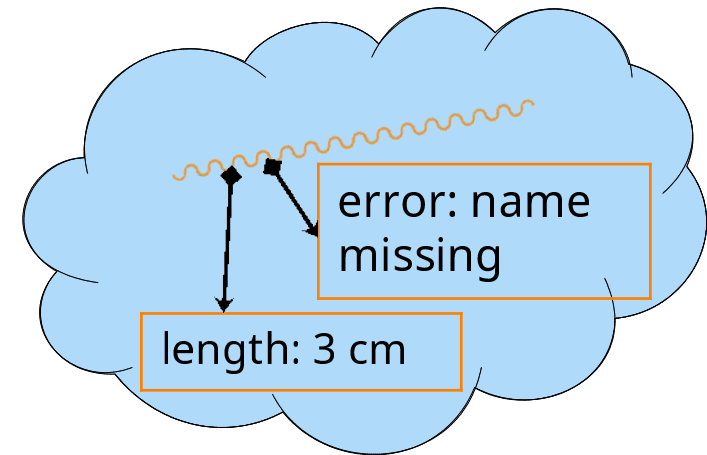
- Findings
- Interpretation results

Solution: LionWeb derived models



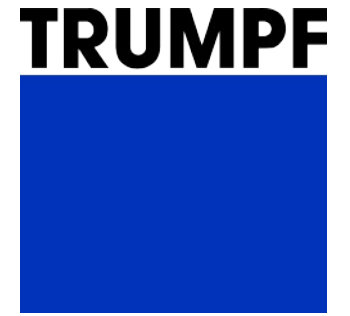
- Temporary solutions
 - Annotations
 - Special care during (de)serialization
 - Reuse serialization, use special channel

original model with annotations

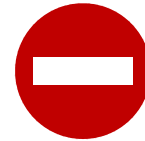




Experience



- DSL for unifying terms / semantic works
- Spikes for trials and vision
- LionWeb infrastructure
- Obvious where "state" lives
- JSON for communication, readable keys helpful (debugging) ...
- Fast implementation of new features ...



- Tight coupling
- Renaming issues → Migration
- Learning effort (language + language engineering)
- Small community
- ... Inefficient, debugging later less needed
→ Binary formats?
- ... after infrastructure has been established

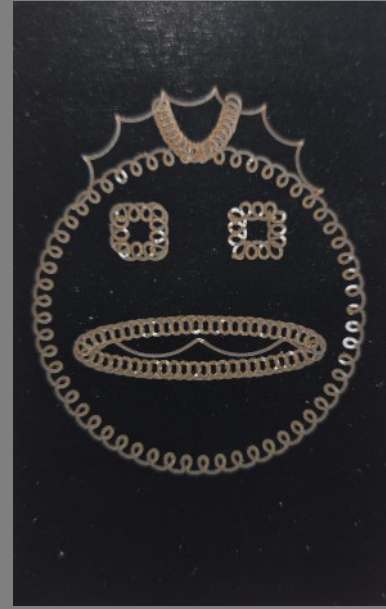
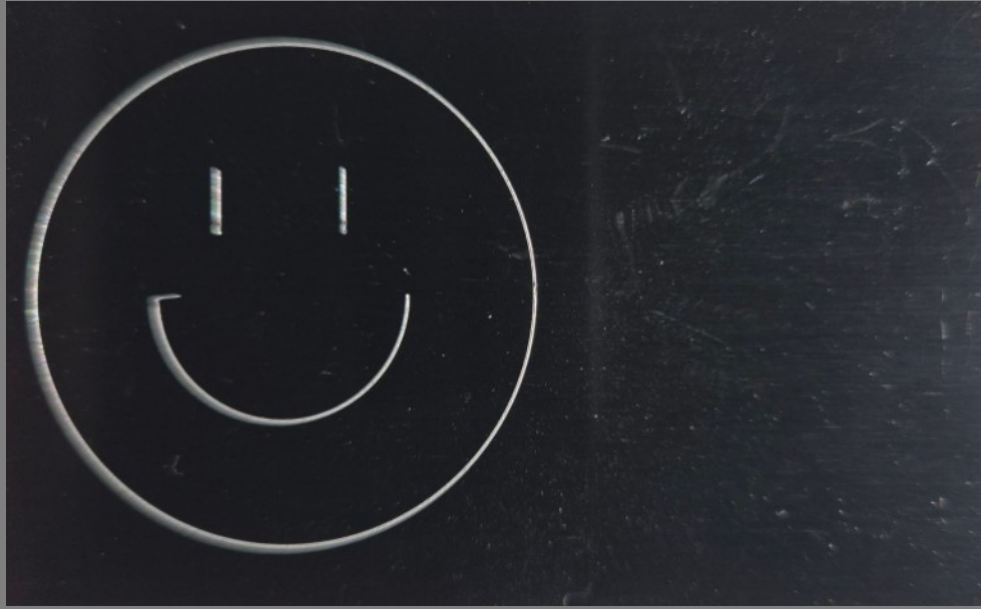




Image attribution

- Browser icon by [Picol](#)