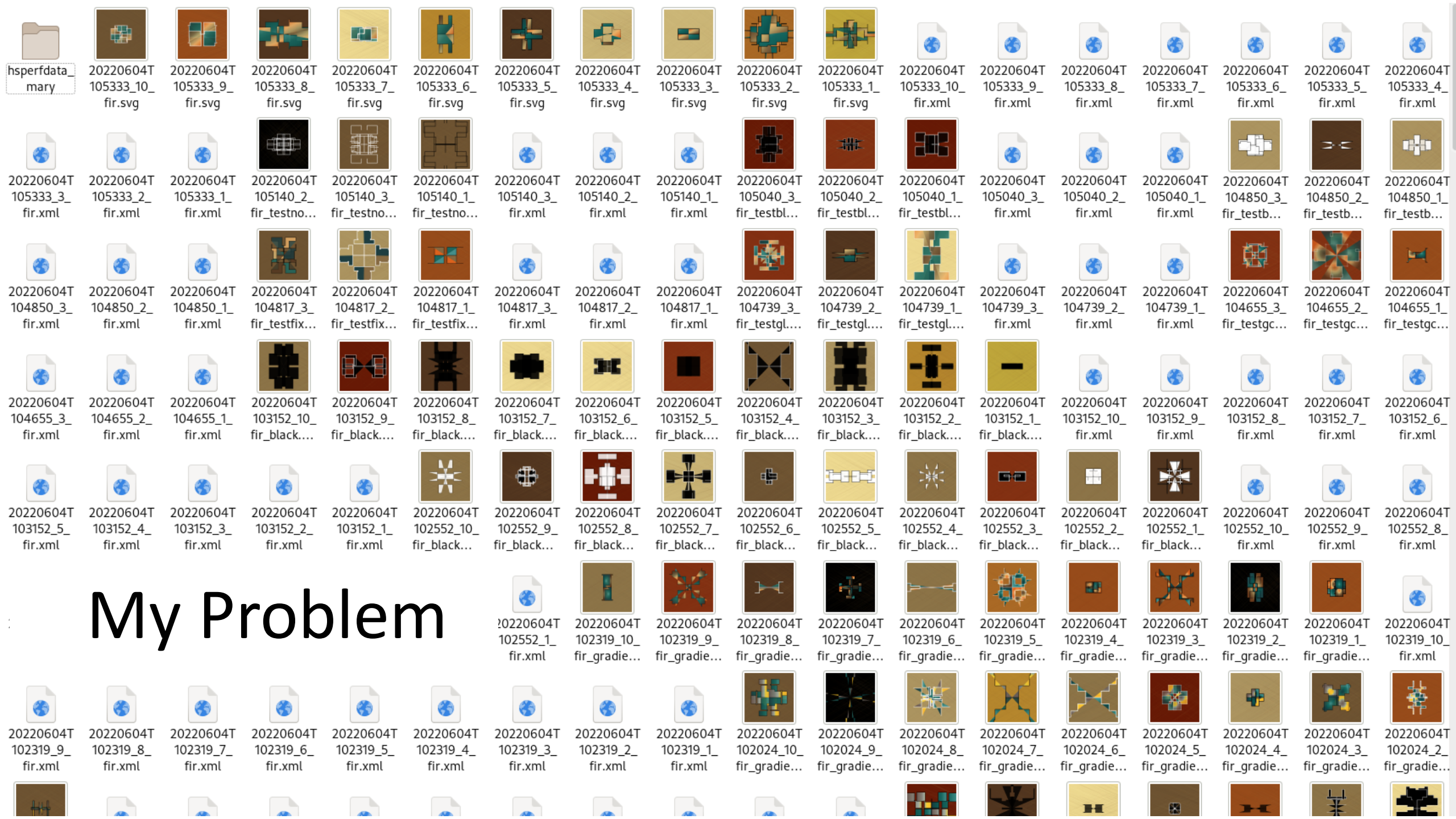




Metadata for Creators

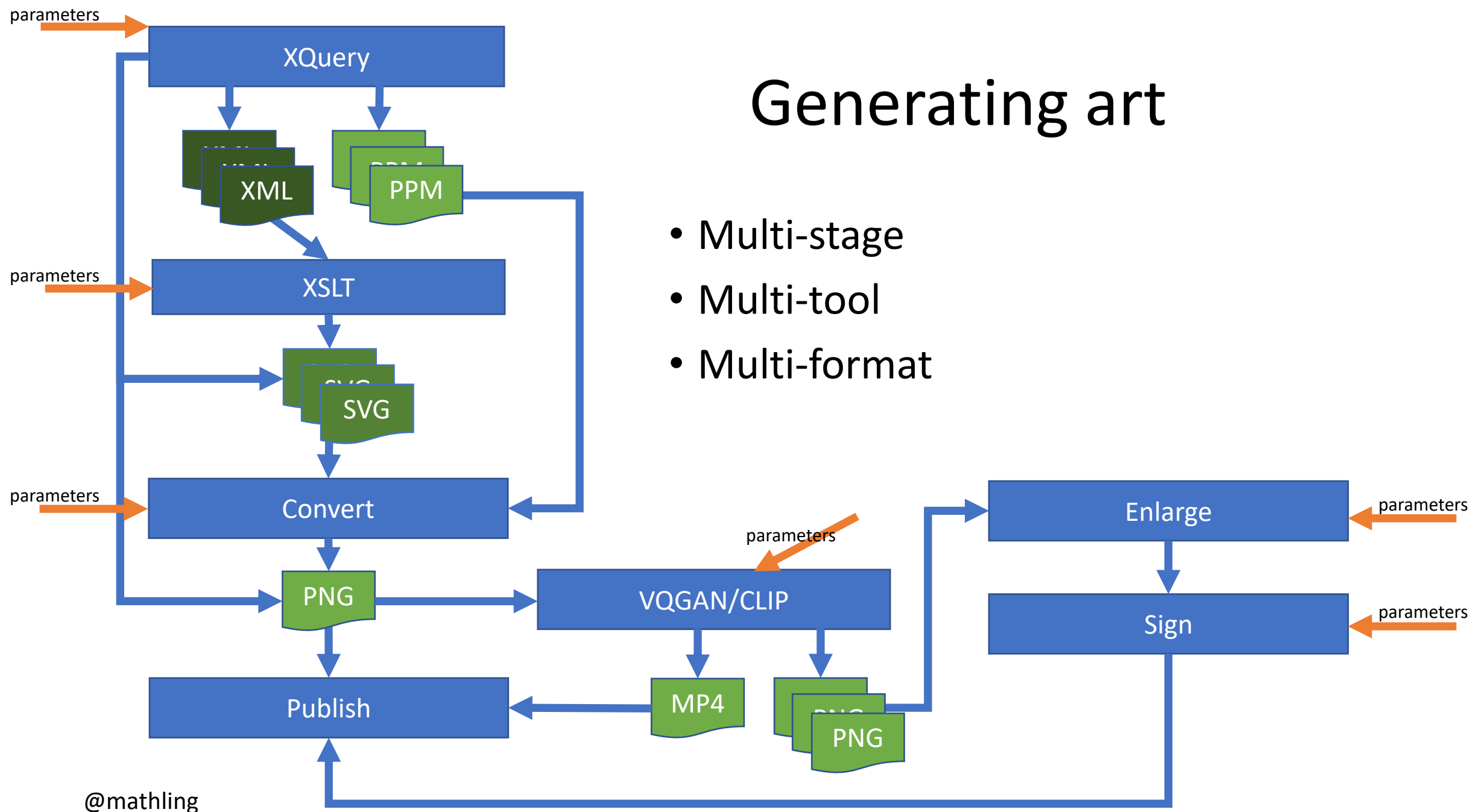
Mary Holstege
@mathling

```
speckles.orientation=inner outer  
speckles.mode=consistent  
speckles.shape=superrose
```



Generating art

- Multi-stage
- Multi-tool
- Multi-format

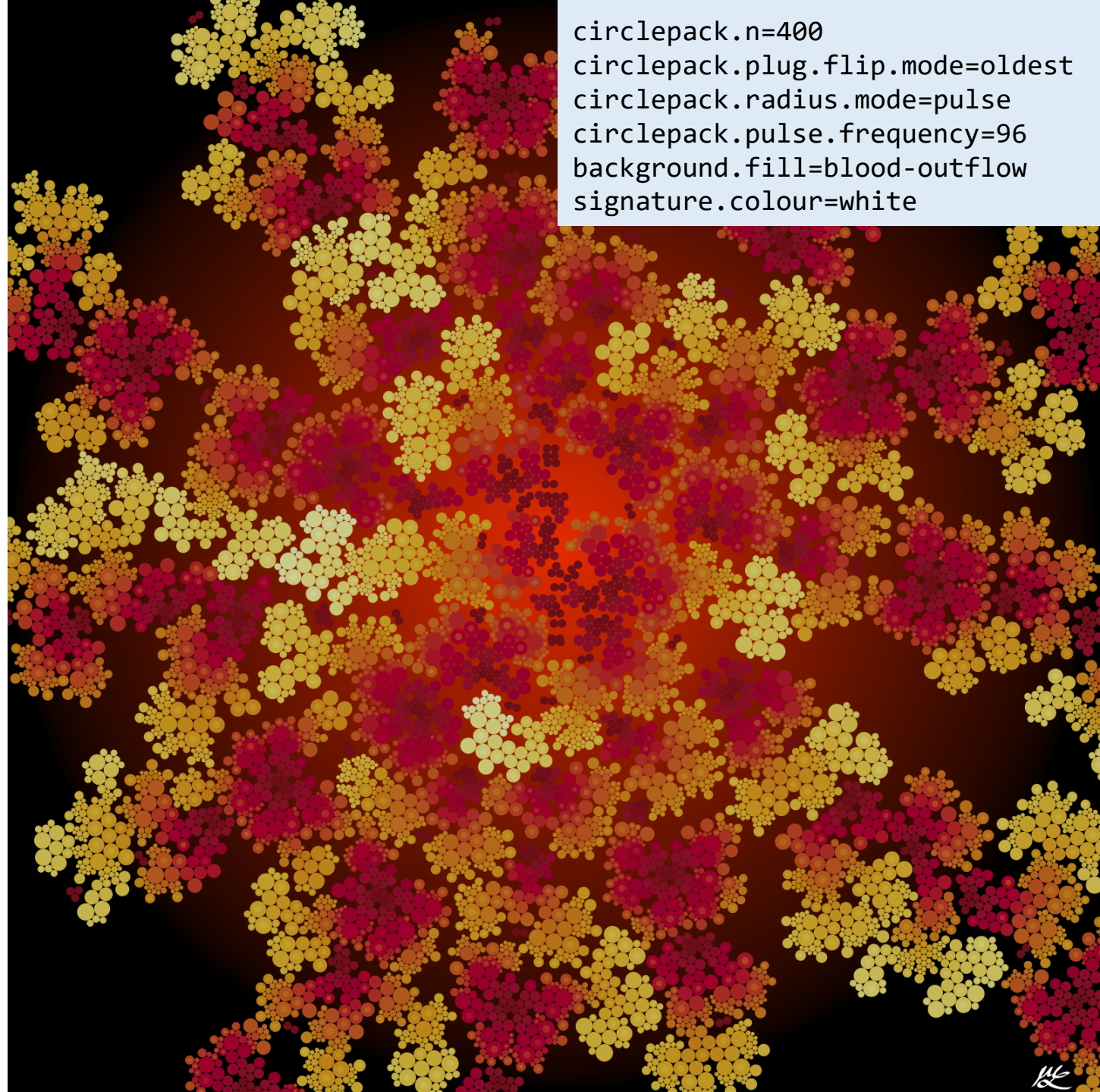


Background

Where are we headed?

@mathling

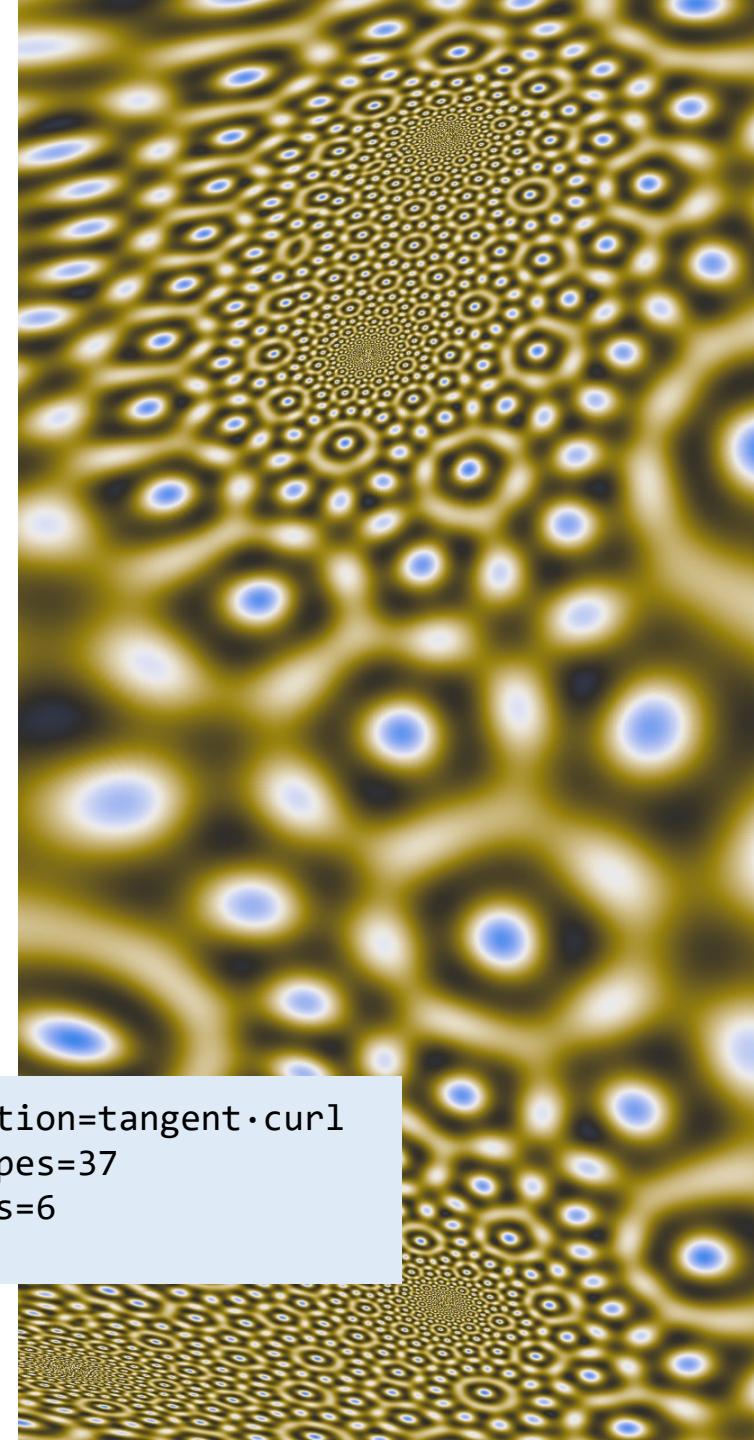
```
circlepack.n=400  
circlepack.plugin.flip.mode=oldest  
circlepack.radius.mode=pulse  
circlepack.pulse.frequency=96  
background.fill=blood-outflow  
signature.colour=white
```



Where are we headed?

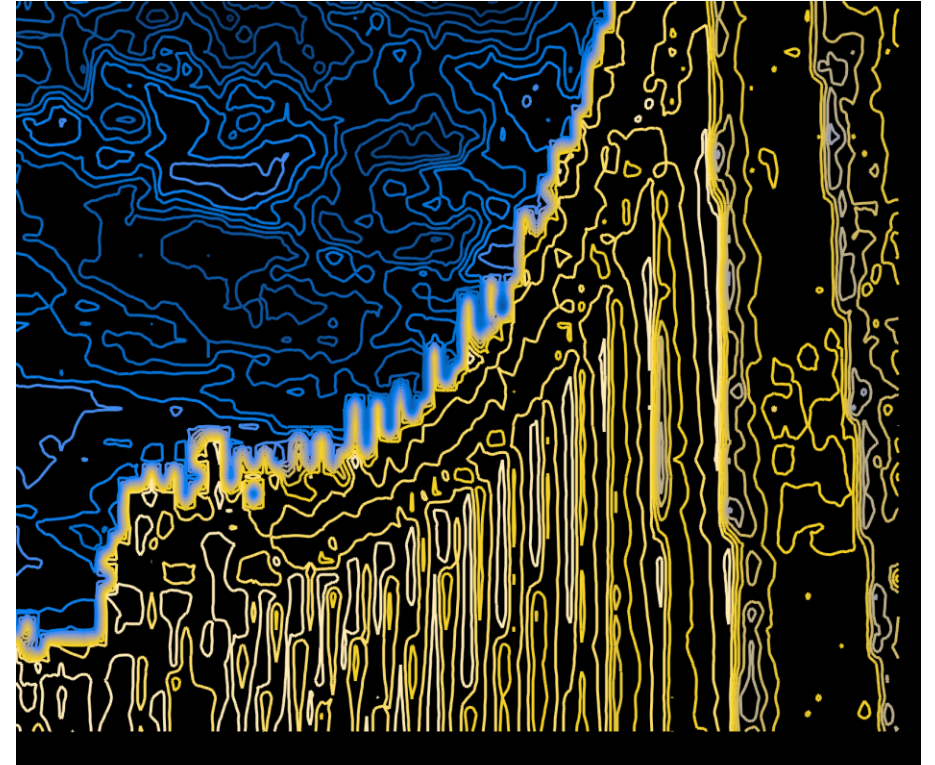
- Metadata as part of the creative process
- Selected lessons on capturing metadata
- **NOT:** code, see the paper

```
quasi.mutation=tangent.curl  
quasi.stripes=37  
quasi.waves=6  
quasi.φ=20
```



What is metadata?

What and who is it for?



```
contourimage.levels=20  
contourimage.granularity=45  
contourimage.gradient=CET-CBL2-reverse  
contourimage.map=xyz  
contourimage.density=3.43
```

The usual metadata story

- Metadata is data about data
- There are three kinds of metadata:
 - Descriptive
 - Administrative
 - Technical
 - Preservation
 - Rights
 - Structural

```
wordplay.stem.gradient=lake  
wordplay.flower.gradient=werner_oranges  
wordplay.font=ShadowsIntoLight-Regular
```



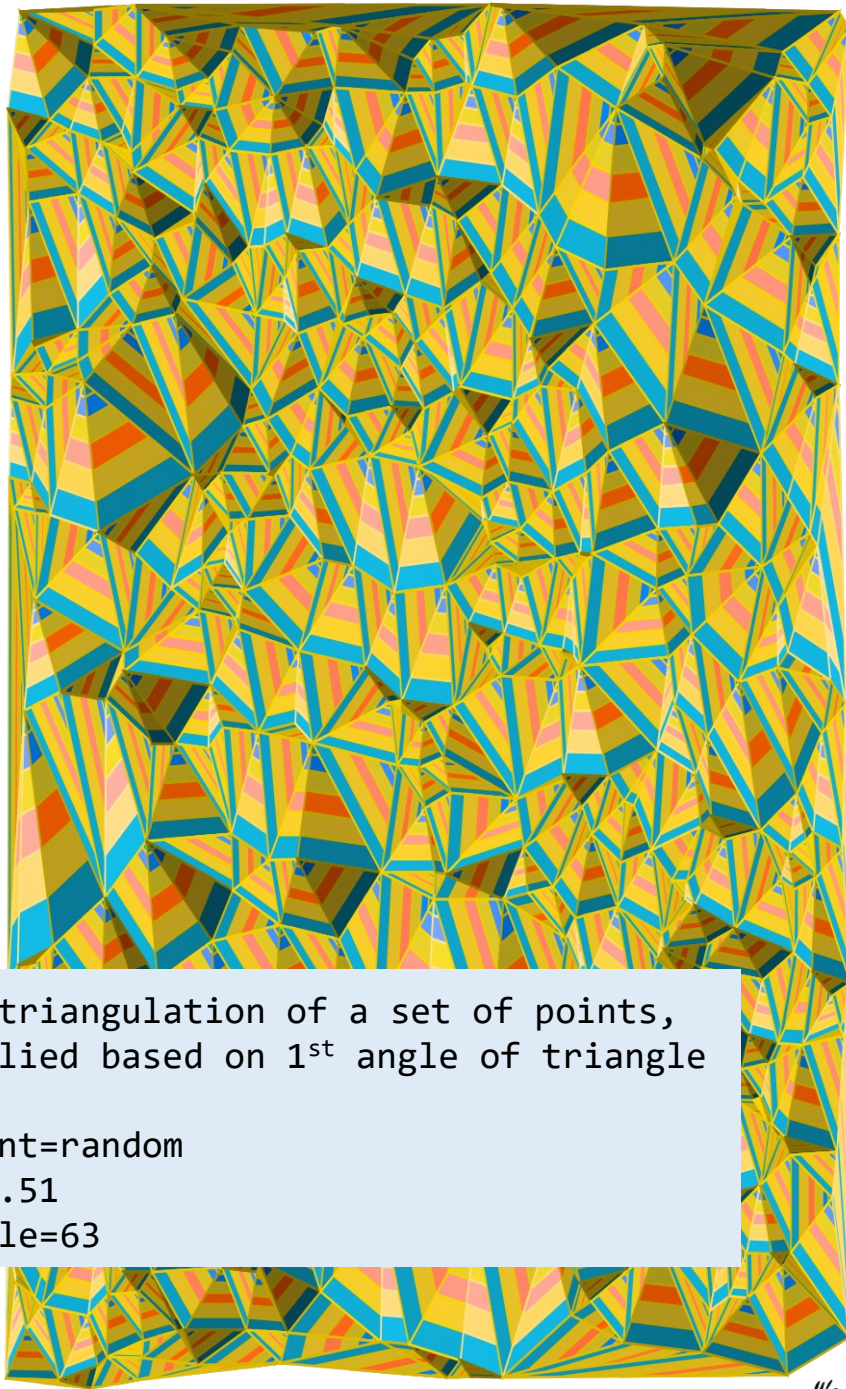
3rd party The usual metadata story

- Metadata is data about data
- There are three kinds of metadata:
 - Descriptive
 - Administrative
 - Technical
 - Preservation
- By archivists and librarians
- Separated from act of creation
- Concerned about standards

```
wordplay.stem.gradient=lake  
wordplay.flower.gradient=werner_oranges  
wordplay.font=ShadowsIntoLight-Regular
```

Metadata for (and by) creators

- As part of the creative process
- For the creator:
 - To communicate with their audience
 - To organize a collection of their works
 - To create new works related to previous works



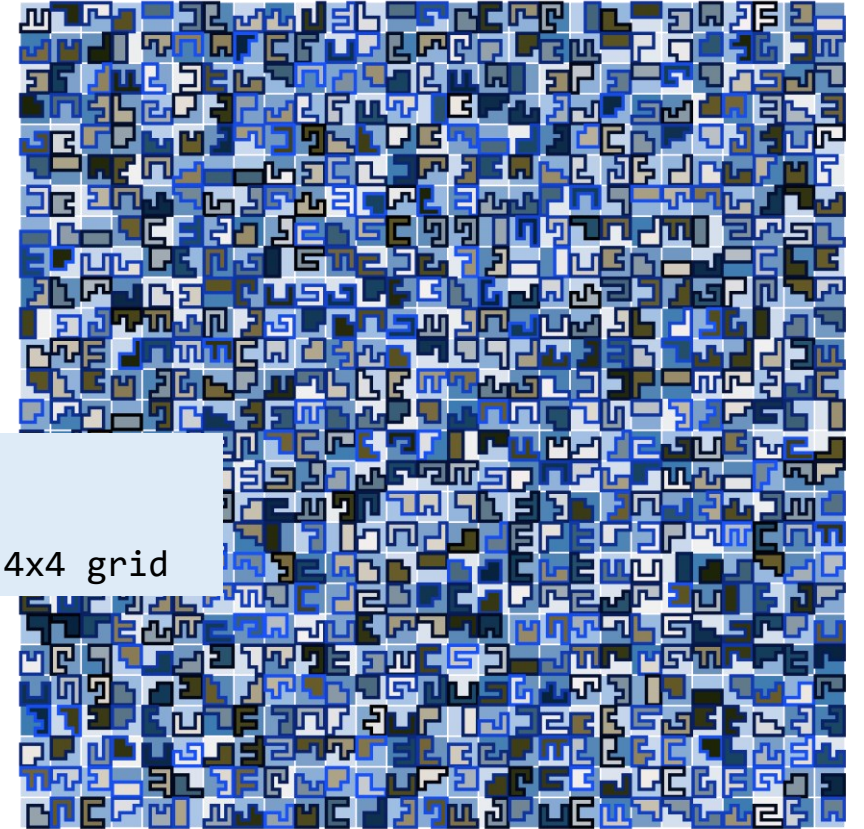
```
colophon=Delaunay triangulation of a set of points,  
lighting model applied based on 1st angle of triangle
```

```
delaunay.arrangement=random  
delaunay.shading=0.51  
delaunay.shade.angle=63
```

Signature metadata

- About the work or creator's relation to it
- Why? Communicate with audience

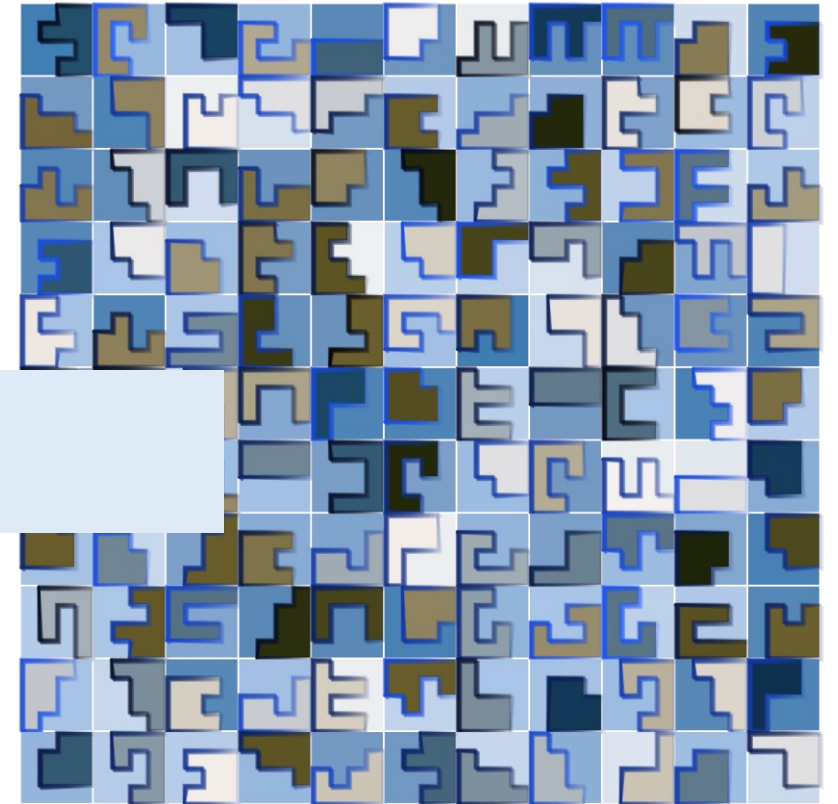
copyright=Copyright© Mary Holstege 2021
description=G4 sea/sky/deepsea
colophon=Tiling based on equiarea divisions of 4x4 grid



Organizational metadata

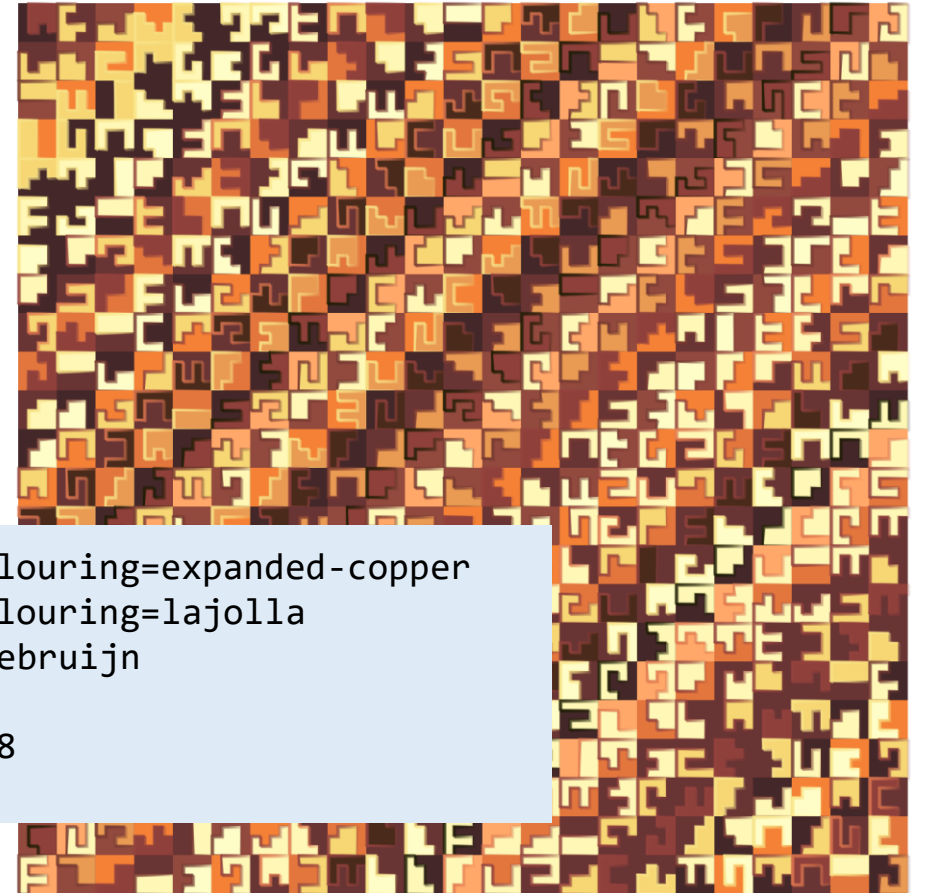
- Assertions about the work and its attributes
- Why? Retrieval or analysis of the work

```
resolution=medium  
background.fill=None  
g4.stroked=True
```



Process metadata

- Assertions about process producing the work
- Why? Enable creations based on this work
- Subtypes:
 - Fixed: Facts true of every output of process
 - Variable: Facts true for a specific output



```
g4.background.colouring=expanded-copper  
g4.foreground.colouring=lajolla  
g4.colour.mode=debruijn  
g4.divisions=25  
g4.fatness=0.7798
```

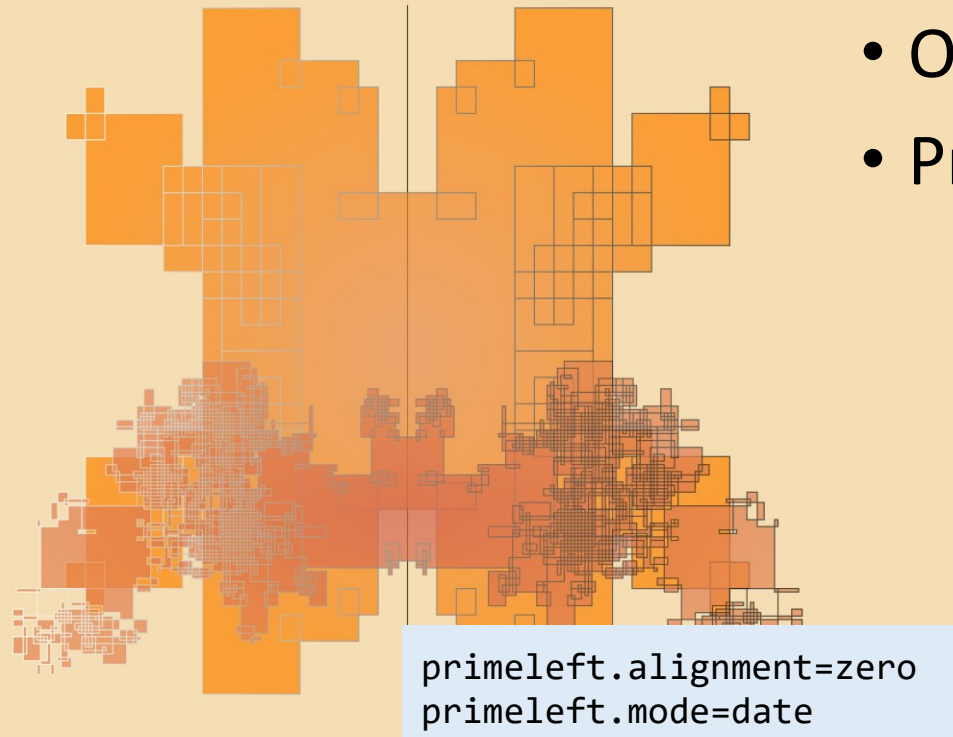
Summary

3rd party: archivists and librarians

- Descriptive
- Administrative
 - Technical
 - Preservation
 - Rights
- Structural

1st party: creators

- Signature
- Organizational
- Process
 - Fixed
 - Variable



Standardization gets in the way, mostly

- “Standardization” as an agreement with future self
- Fluid and emergent, as variable as processes
- External standards relevant only for publication (kinda maybe not)

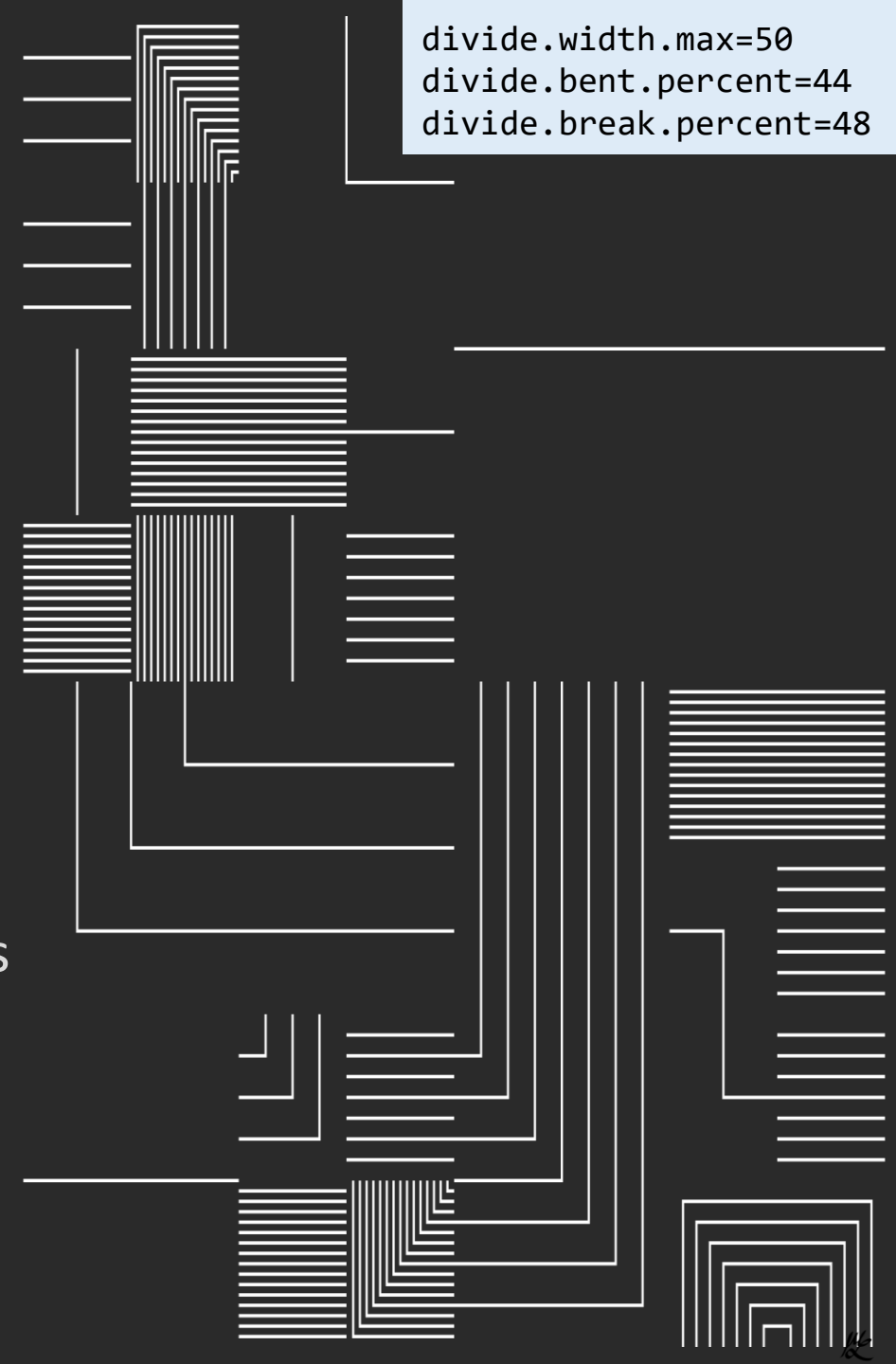


dc:Creator=Mary Holstege
dc:Description=Multi-level triangular Truchet tiling, with VQGAN+CLIP applied for a few iterations
dc:Rights=Copyright© Mary Holstege 2021

Some lessons

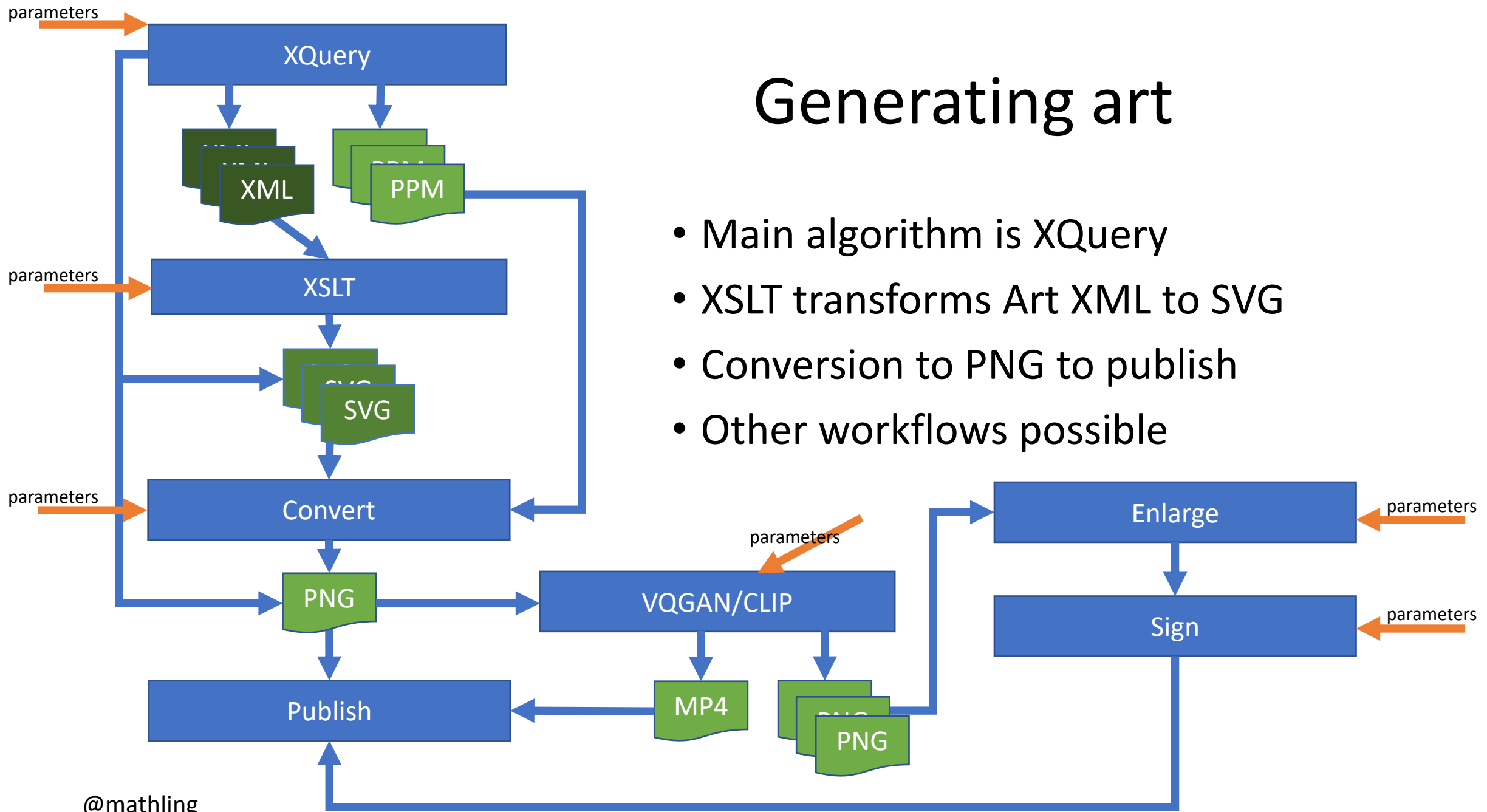
Make generating metadata as easy as generating works

@mathling



Generating art

- Main algorithm is XQuery
- XSLT transforms Art XML to SVG
- Conversion to PNG to publish
- Other workflows possible



Embed all metadata

- What is separated can become lost
- In my case
 - Metadata elements in XML
 - Comment/metadata fields in images



Copyright © Mary Holstege 2020.

Visualization of the Exoplanet catalog (<https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbIs&config=planets>)

Arc mapping: ordering=pl_disc,pl_publ_date,rowupdate arc-length:pl_rade colour:pl_orbper width:pl_bmasse opacity:pl_st_npar

Stars mapping: direction:ra distance:st_dist spokes:pl_pnum arc-length:st_rad colour:st_teff opacity:st_optmag

Use mean for missing data

Map log of orbital period and mass because of very wide range

Star colour mapping is linear in 3 distinct ranges to get better spread around mode

Automate (as much as possible)

- People are lazy and forgetful
- That which is manual is more likely to be wrong
- Everything starts as a special case
- In my case:
 - Framework dumps out parameter tables
 - Stylesheets preserve/extend metadata
 - Conversion scripts preserve metadata

```
planet.scaling=400  
planet.background=rain-outflow  
planet.swirls=navy  
sun.colour=rgb(245,252,100)  
sun.n-rays=200
```



Everything important needs a name



- Link output to process
 - Easier to comprehend
 - Naming **systems**
-
- In my case:
 - Names, names, names
 - Libraries use naming systems to create aspects of the output

Use descriptors

- Complex/variable pieces of behaviour
 - Can describe for metadata
 - Can execute for effect
- In my case:
 - Descriptors as XQuery maps
 - Functions to interpret and describe them

```
weave.weaving=weaving[56 3 prime(227,22,2) X thue-morse(3)]
```

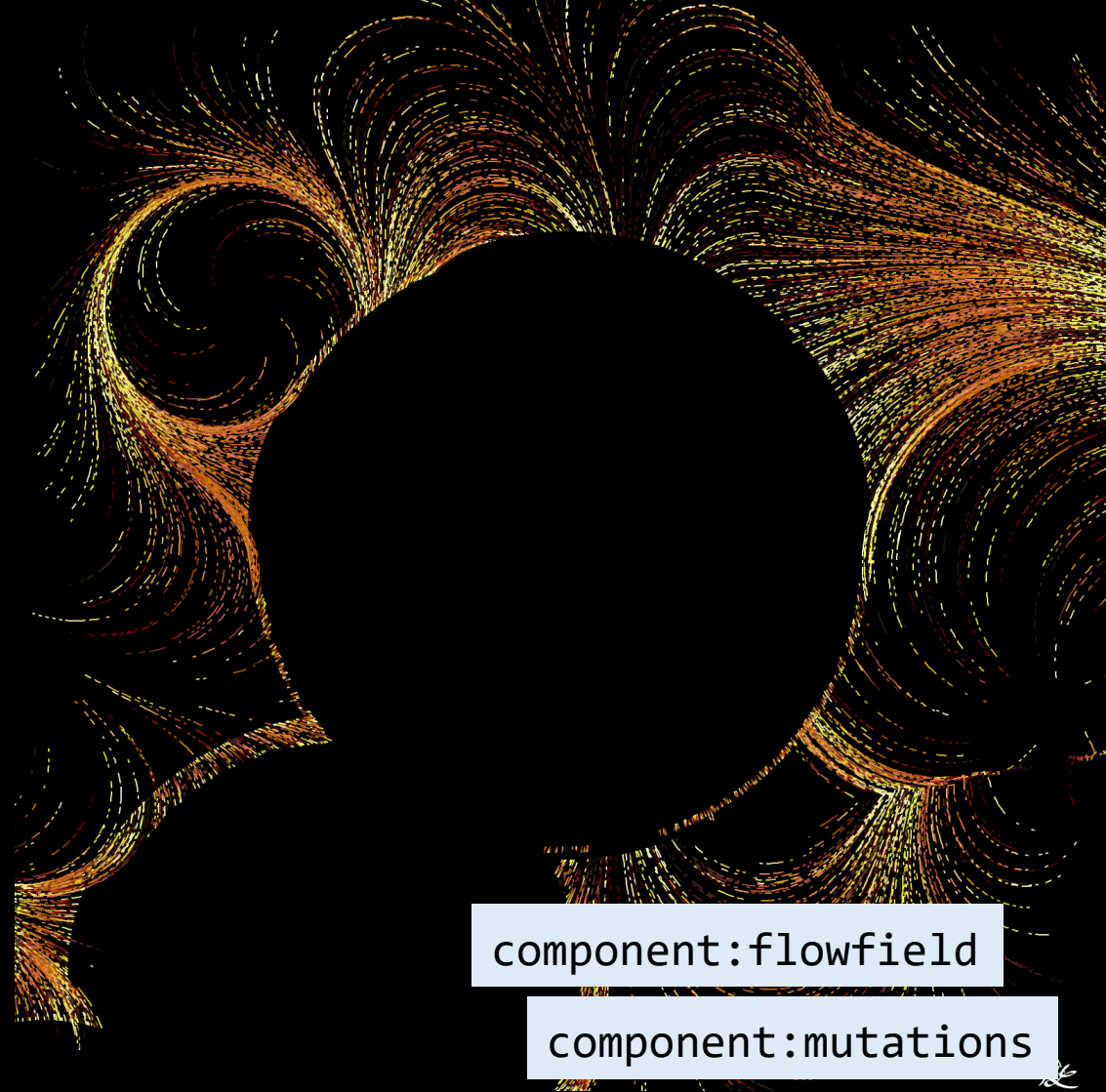


```
map {  
  "kind": "modulated-knot",  
  "p": 3,  
  "x-factors": (  
    map {"q": 7, "r": 0.52, "skew": true()},  
    map {"q": 4, "r": 0.11, "skew": false()}  
  ),  
  "y-factors": (  
    map {"q": 5, "r": -0.9, "skew": false()},  
    map {"q": 4, "r": 0.7, "skew": false()}  
  ),  
  "stretch": 1.25,  
  "openness": 1  
}
```

$$\begin{aligned}x &= 1.25 \cdot \cos(3\theta) * (0.7 \cdot \sin(7\theta) + 0.11 \cdot \cos(4\theta)) \\ y &= \cos(3\theta) * (-0.9 \cdot \cos(5\theta) + 0.7 \cdot \cos(4\theta)) \\ z &= 0.2 \cdot \sin(7\theta)\end{aligned}$$

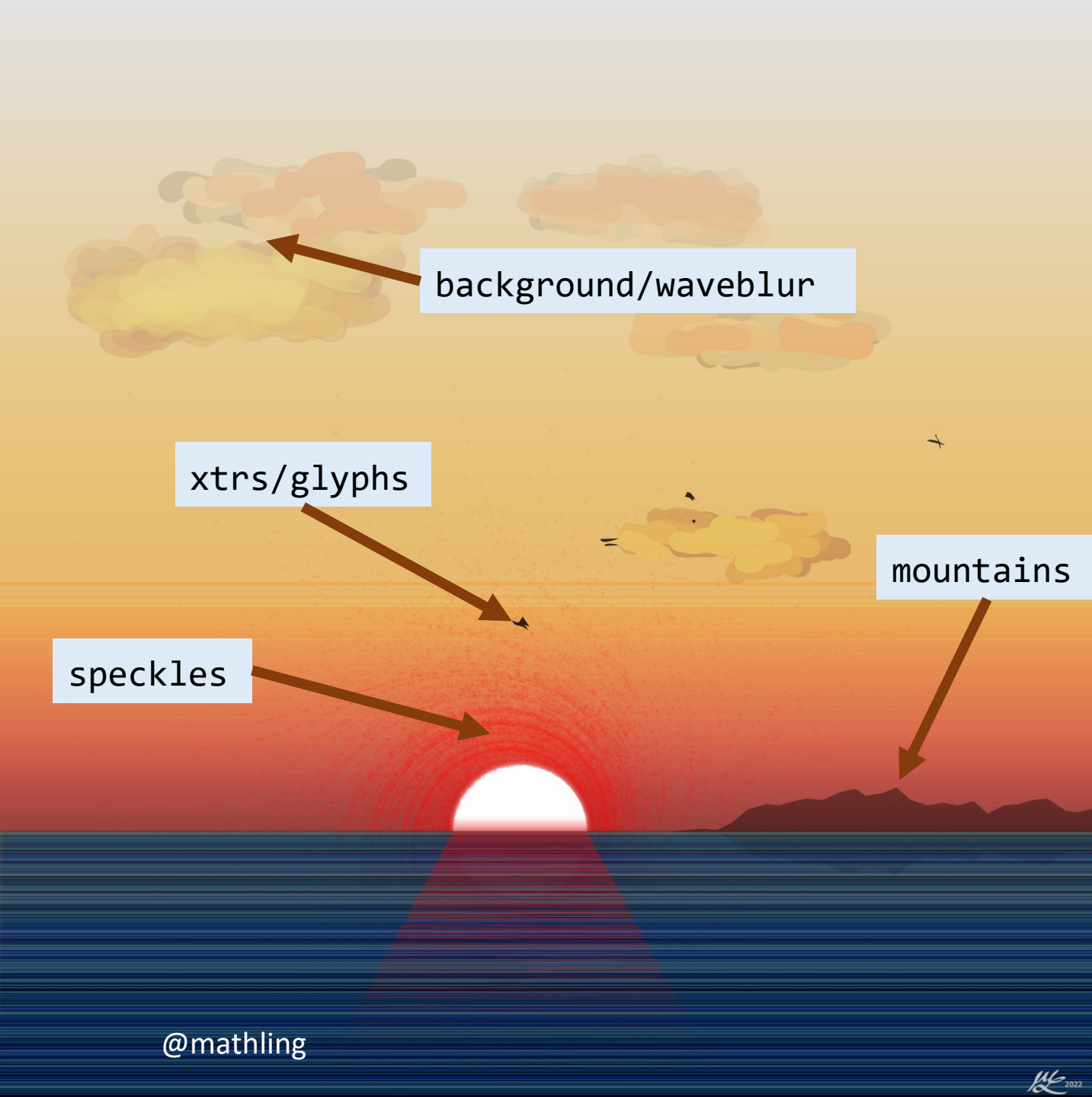
Divide and conquer

- Componentize
 - Component handles its own metadata
- Be selective:
 - Emit metadata for relevant components
- Automate:
 - Component framework
- In my case:
 - Standard metadata callback functions
 - Framework combines named components
 - Recursively
 - Stylesheets introspect on use of component elements



`component:flowfield`

`component:mutations`



background/waveblur

xtrs/glyphs

speckles

mountains

```
component=background  
background.wave.waviness=5  
background.wave.spread=0.7
```

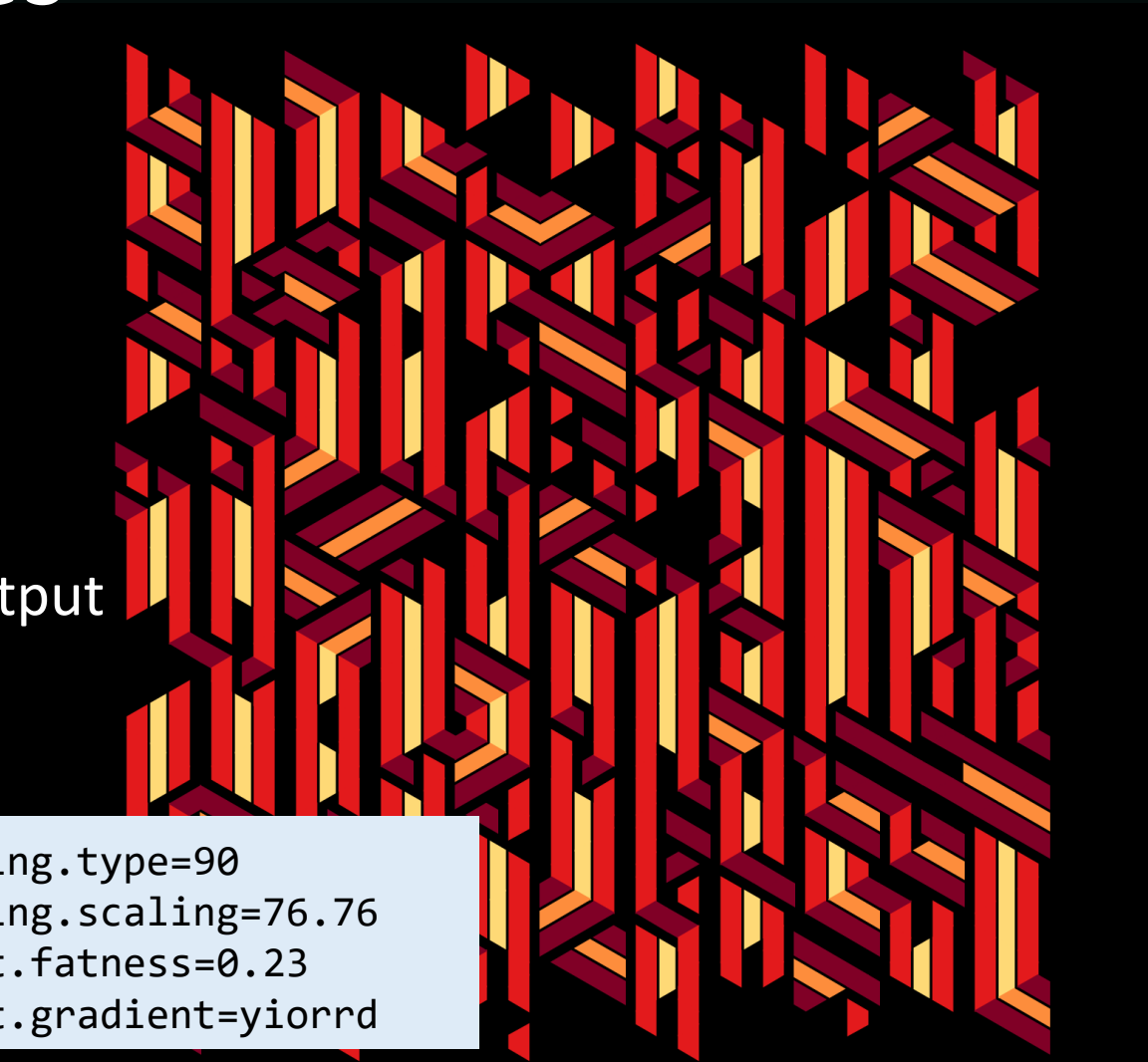
```
component=xtrs  
xtrs.invert=flip(25)
```

```
component=speckles  
speckles.orientation=outer  
speckles.density.multiplier=0.3
```

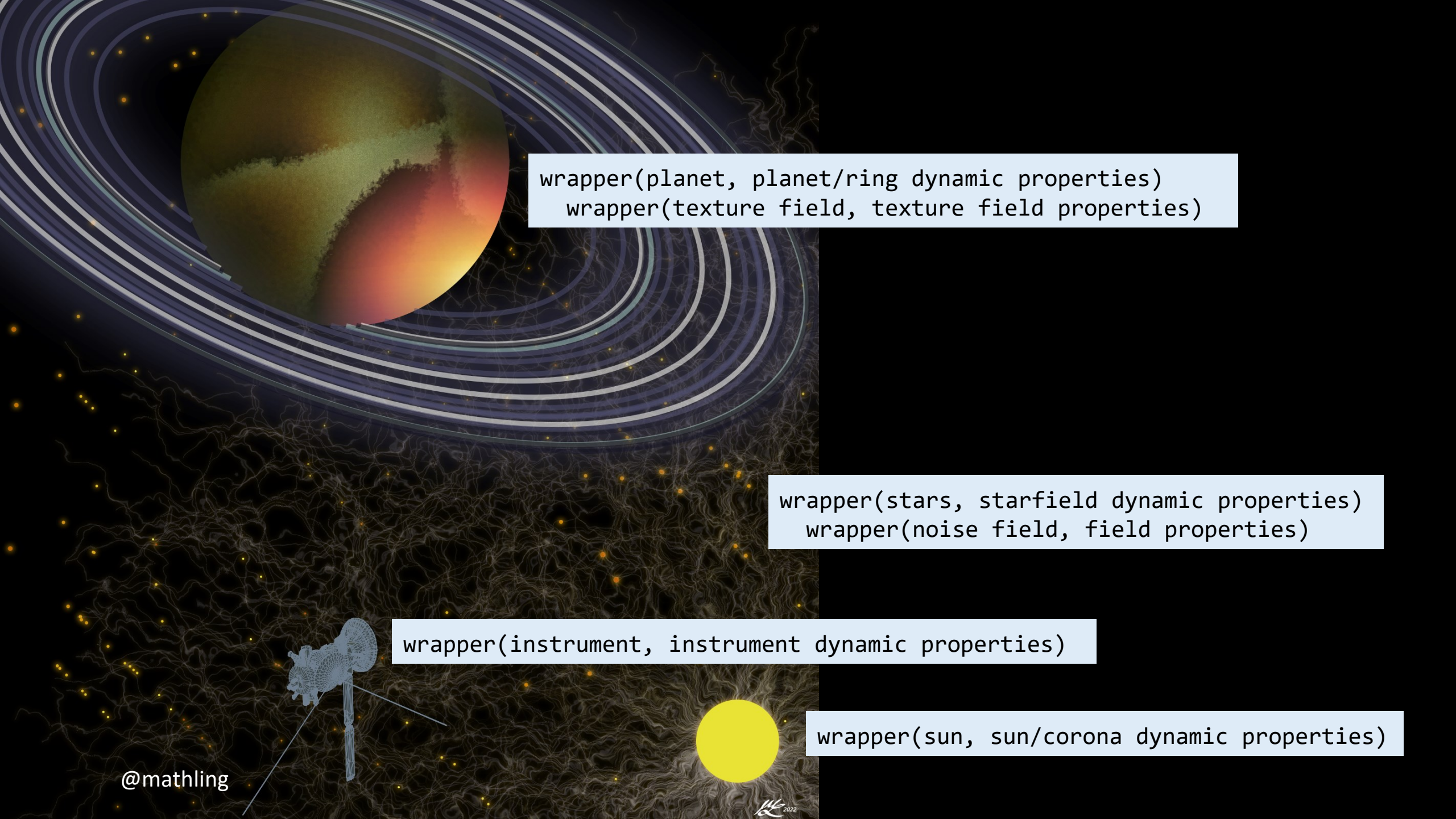
```
component=mountains  
mountain.jags=30  
mountain.spikiness=15
```

Design for dynamic values

- Capture dynamic run time values
- Propagate from nested components
- In my case:
 - Wrappers unite dynamic values with output
 - Components produce wrappers



```
tiling.type=90  
tiling.scaling=76.76  
bent.fatness=0.23  
bent.gradient=yiorrd
```



wrapper(planet, planet/ring dynamic properties)
wrapper(texture field, texture field properties)

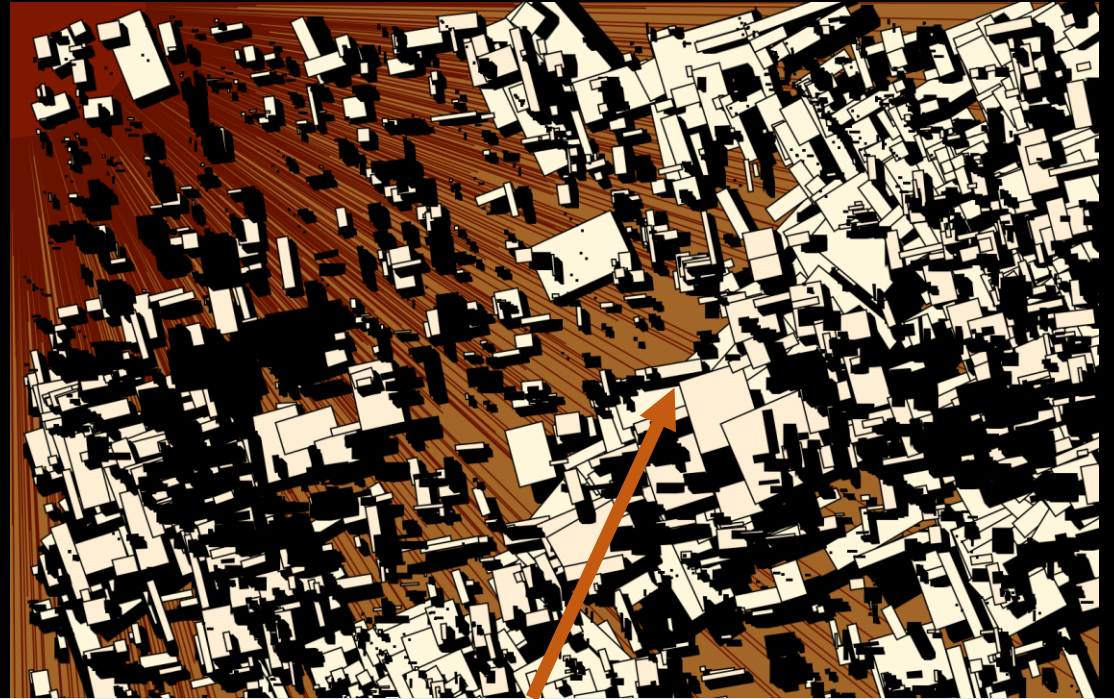
wrapper(stars, starfield dynamic properties)
wrapper(noise field, field properties)

wrapper(instrument, instrument dynamic properties)

wrapper(sun, sun/corona dynamic properties)

Tame complexity with introspection

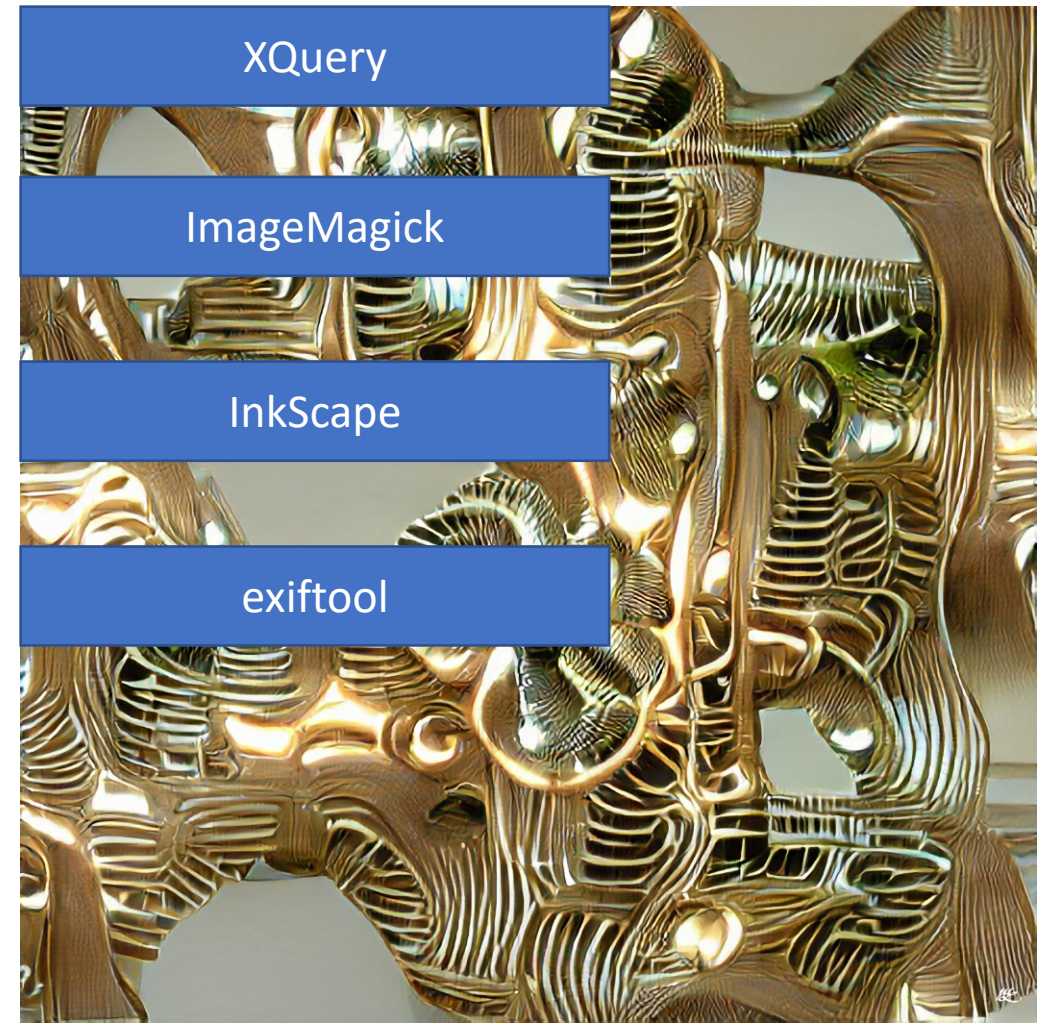
- Some complex behaviours represented as nested function values
- Function annotations and introspection to capture metadata
- In my case:
 - Tricky technique in paper

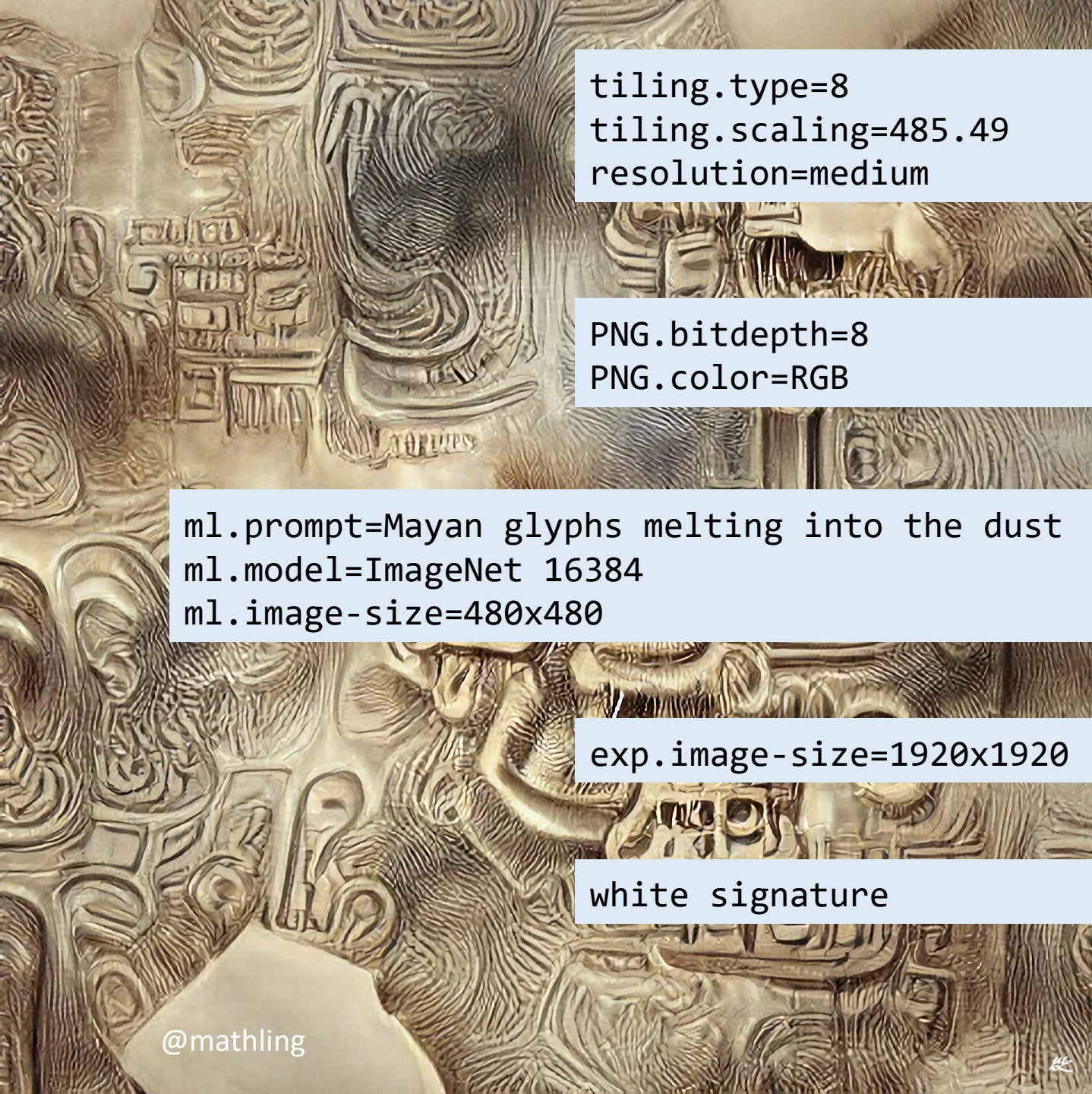


```
mod:scale(  
  mod:abs(ridged:noise({  
    octaves:12 dimension:2 attenuation:2  
    lacunarity:2.09 frequency:1 persistence:1.5  
  })),  
  mod:constant(0.000625), mod:constant(0.000625))
```


Pay attention to the whole publication chain

- Every step has its own metadata to add
- Format changes apt to lose metadata
- **Many** tools bad at metadata
- In my case:
 - Several tools combine to preserve metadata





```
tiling.type=8  
tiling.scaling=485.49  
resolution=medium
```

XQuery

```
PNG.bitdepth=8  
PNG.color=RGB
```

Convert

```
ml.prompt=Mayan glyphs melting into the dust  
ml.model=ImageNet 16384  
ml.image-size=480x480
```

VQGAN/CLIP

```
exp.image-size=1920x1920
```

Enlarge

```
white signature
```

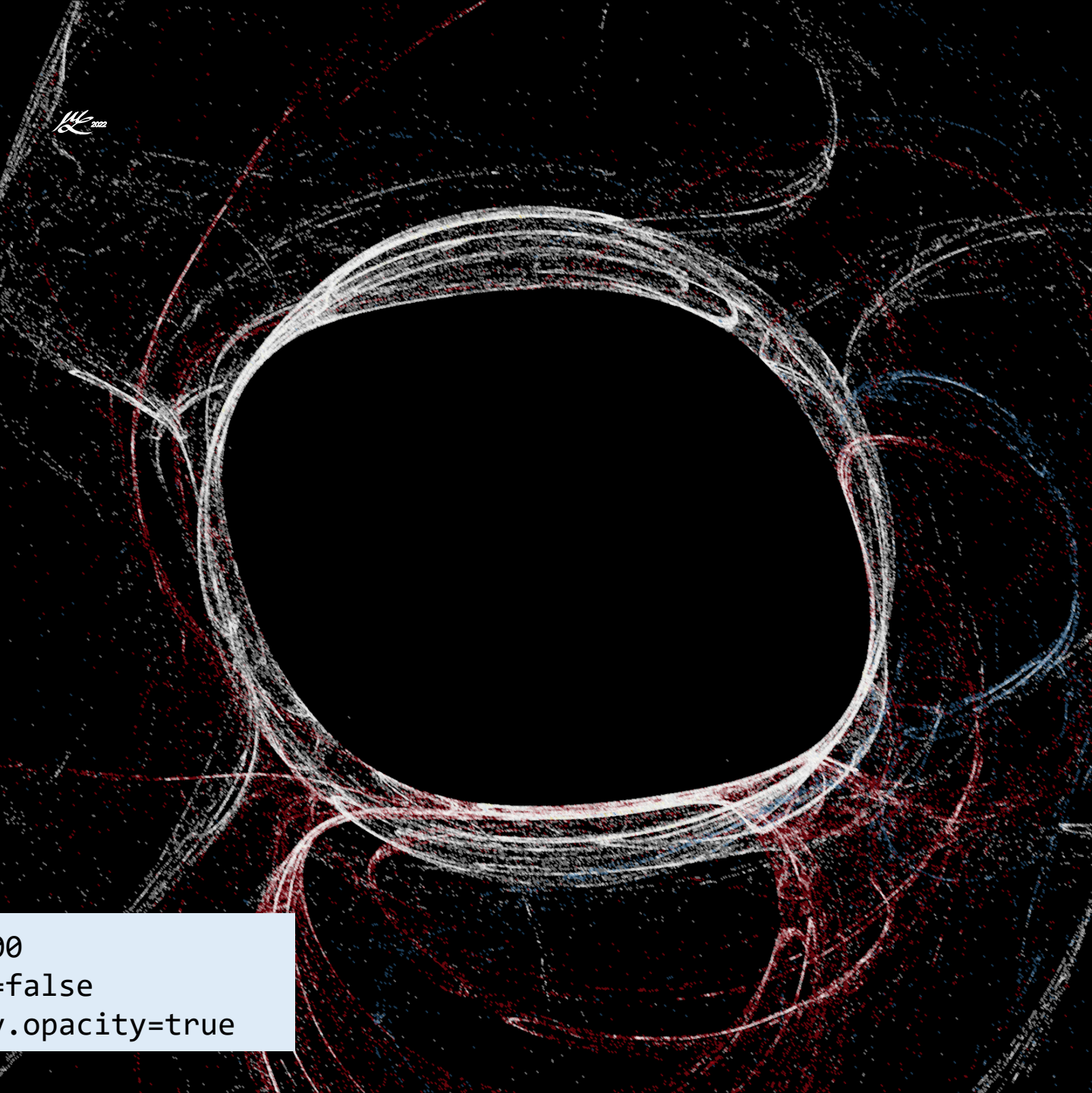
Sign

Recap

Main points

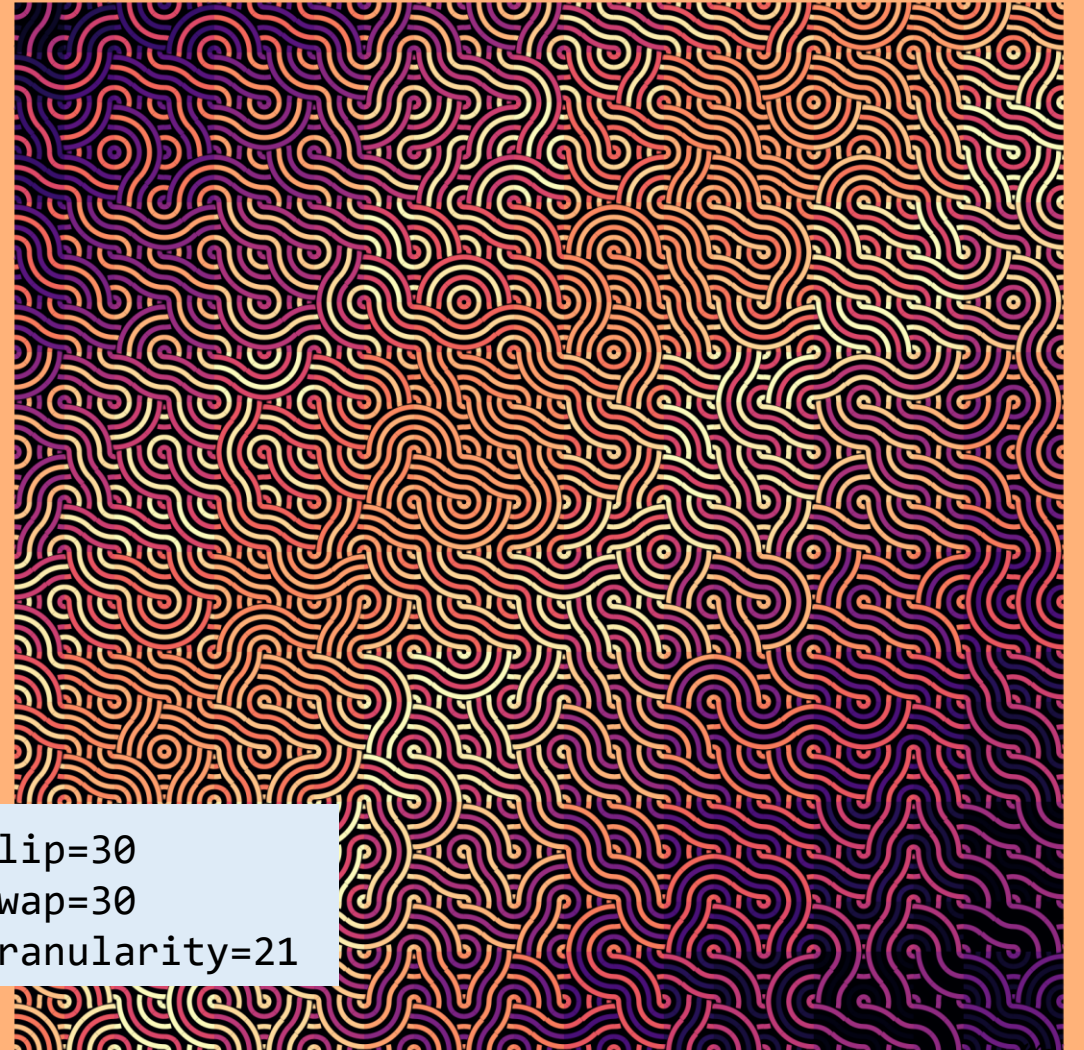
@mathling

```
f Flame.n=500000  
f Flame.smooth=false  
f Flame.density.opacity=true
```



Metadata by and for creators

- 1st party perspective
- Metadata for...
 - Communicating with audience
 - Finding and analyzing works
 - Creating new works from old
- Consistency, not standardization



```
swirls.flip=30  
swirls.swap=30  
swirls.granularity=21
```


Metadata-first development

- Make metadata creation as easy as data creation
- Positive feedback loop
 - Embedding enables reuse
 - Automation begets automation
 - Metadata enablers enable implementation too
- Don't shy away from complexity

```
divide.k.max=3  
divide.bent.percent=65  
divide.break.percent=89
```



Discussion

@mathling

```
j23.n-plants=2
j23.plant-1=7@16 budding axiom=?A
rules={
  A:TF[+++X]TFB
  B:TF[---Y]TFA
  C:{ [\\@]:80 [\\\\@]:20 }
  L:[N][M]
  M:[-M{F}FOF}
  N:[+N{F}FOF}
  O:O>F
  T:TF
  X:A
  Y:BC
}
j23.plant-2=4@14 shrub axiom=?F
rules={
  F:{
    FF+[ /+F-F-F ]-[ /-F+F+F ]:80
    FF+[ /-F+F+F ]-[ /+F-F-F ]:20
  }
}
```