

Unifying Model- and Screen Sharing

Yentl Van Tendeloo

Yentl.VanTendeloo@uantwerpen.be

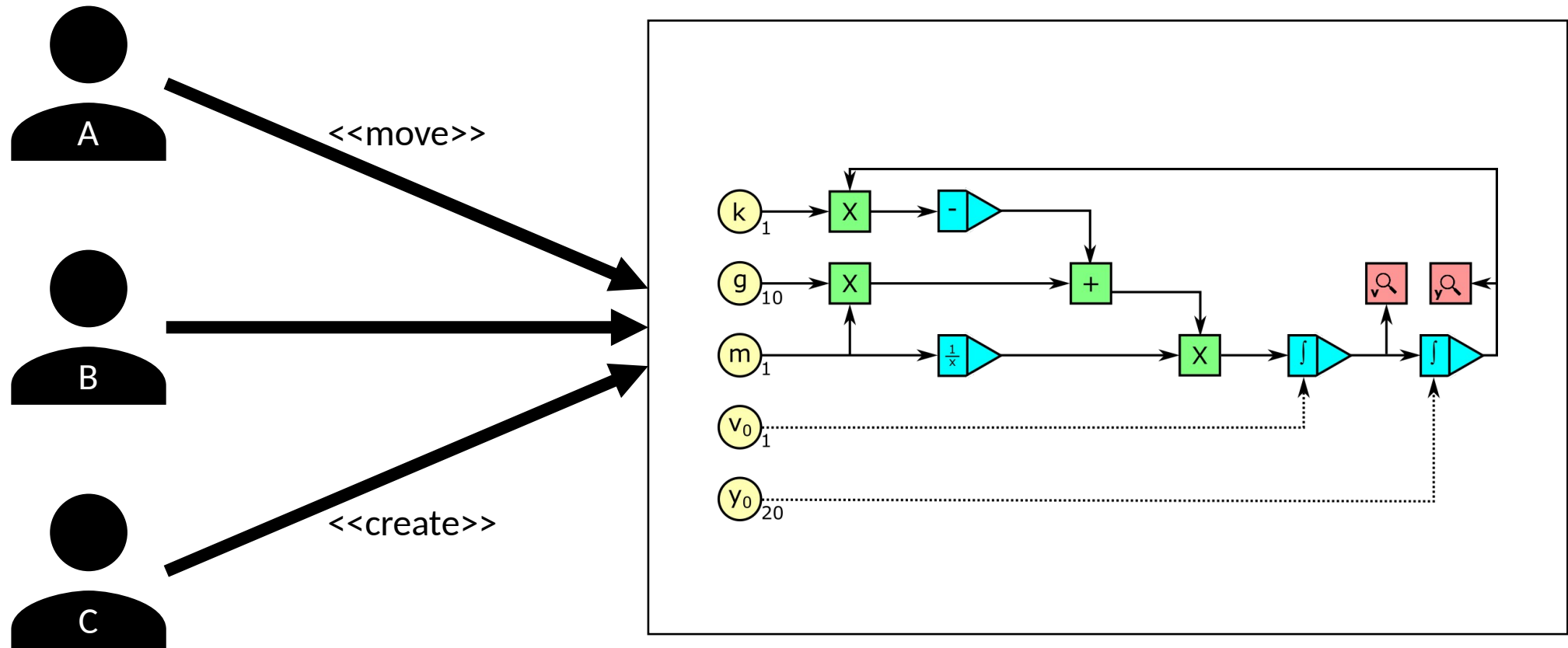
Hans Vangheluwe

Hans.Vangheluwe@uantwerpen.be

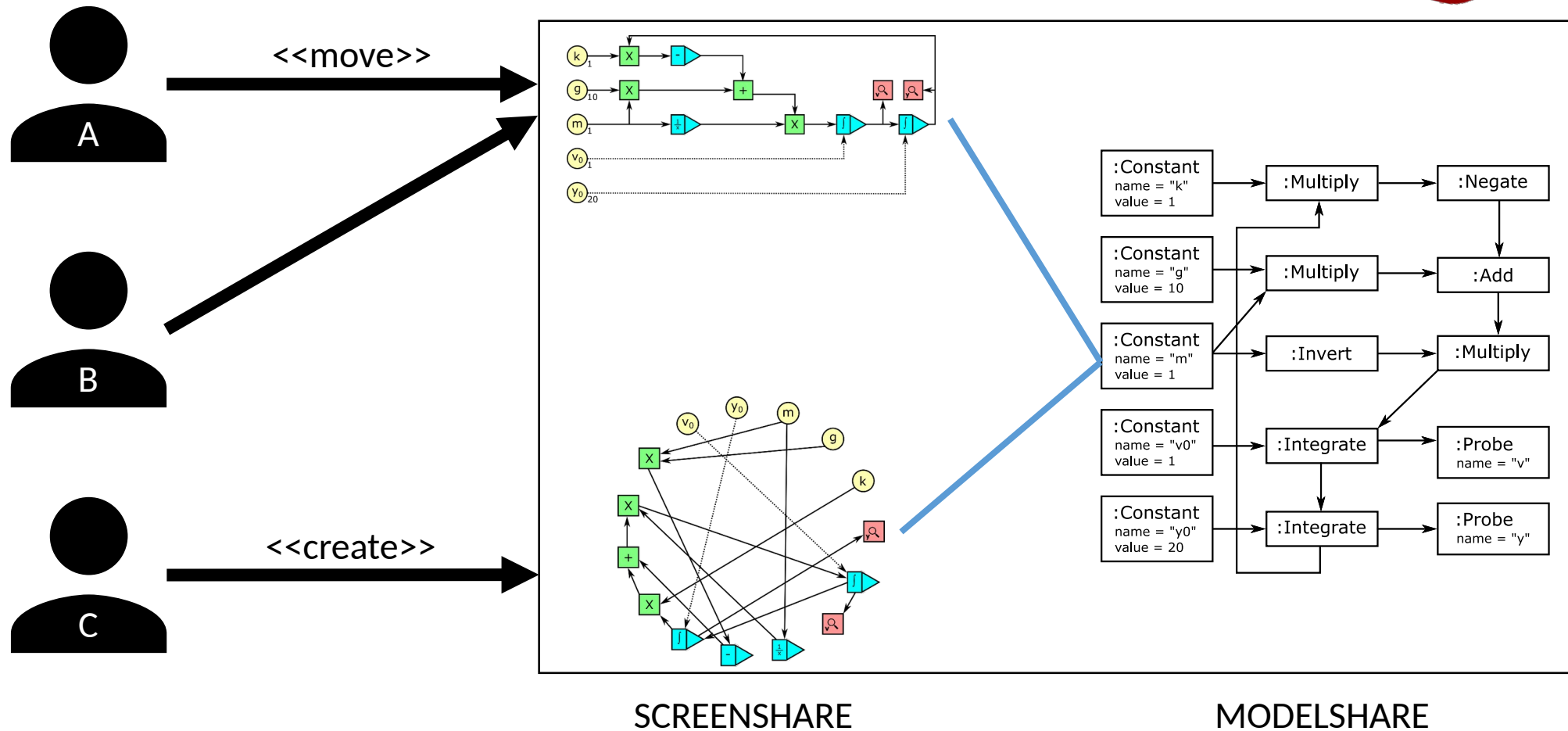


McGill

Collaborative Modelling



Collaborative Modelling



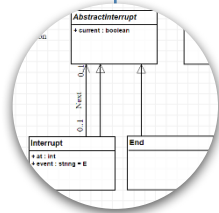
Context



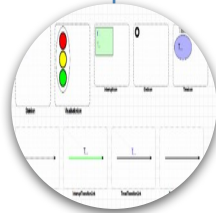
Formalism



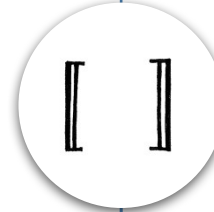
Syntax



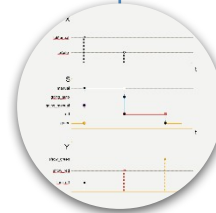
Abstract



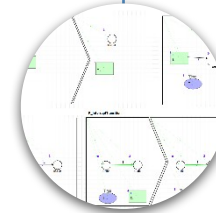
Concrete



Semantics

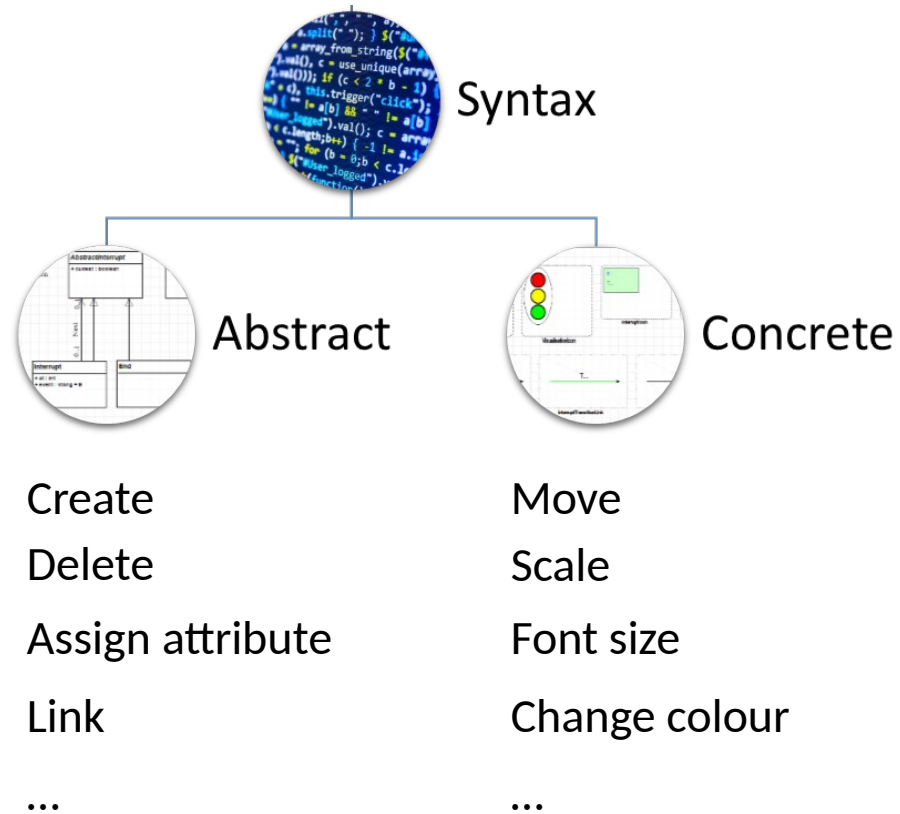


Domain

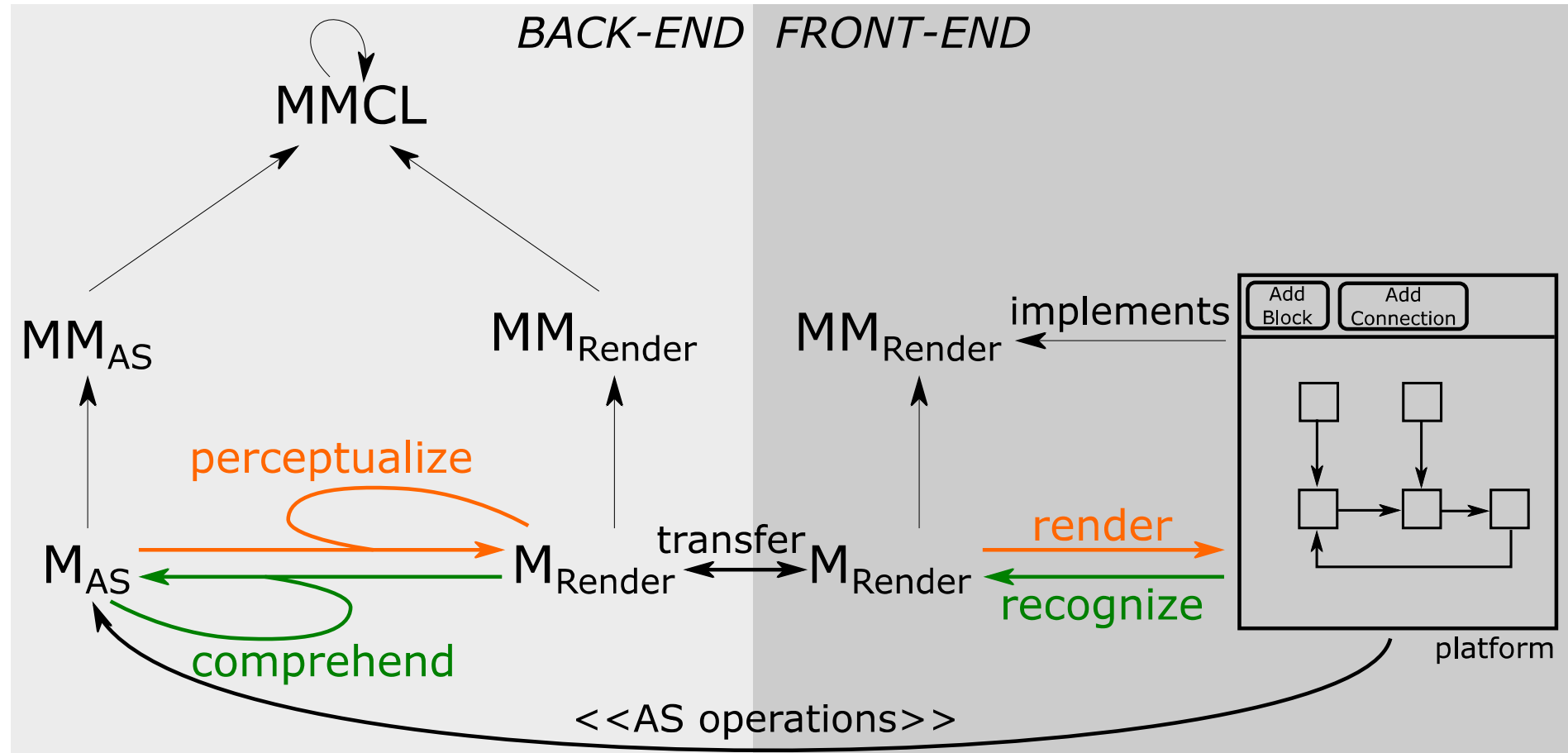


Mapping

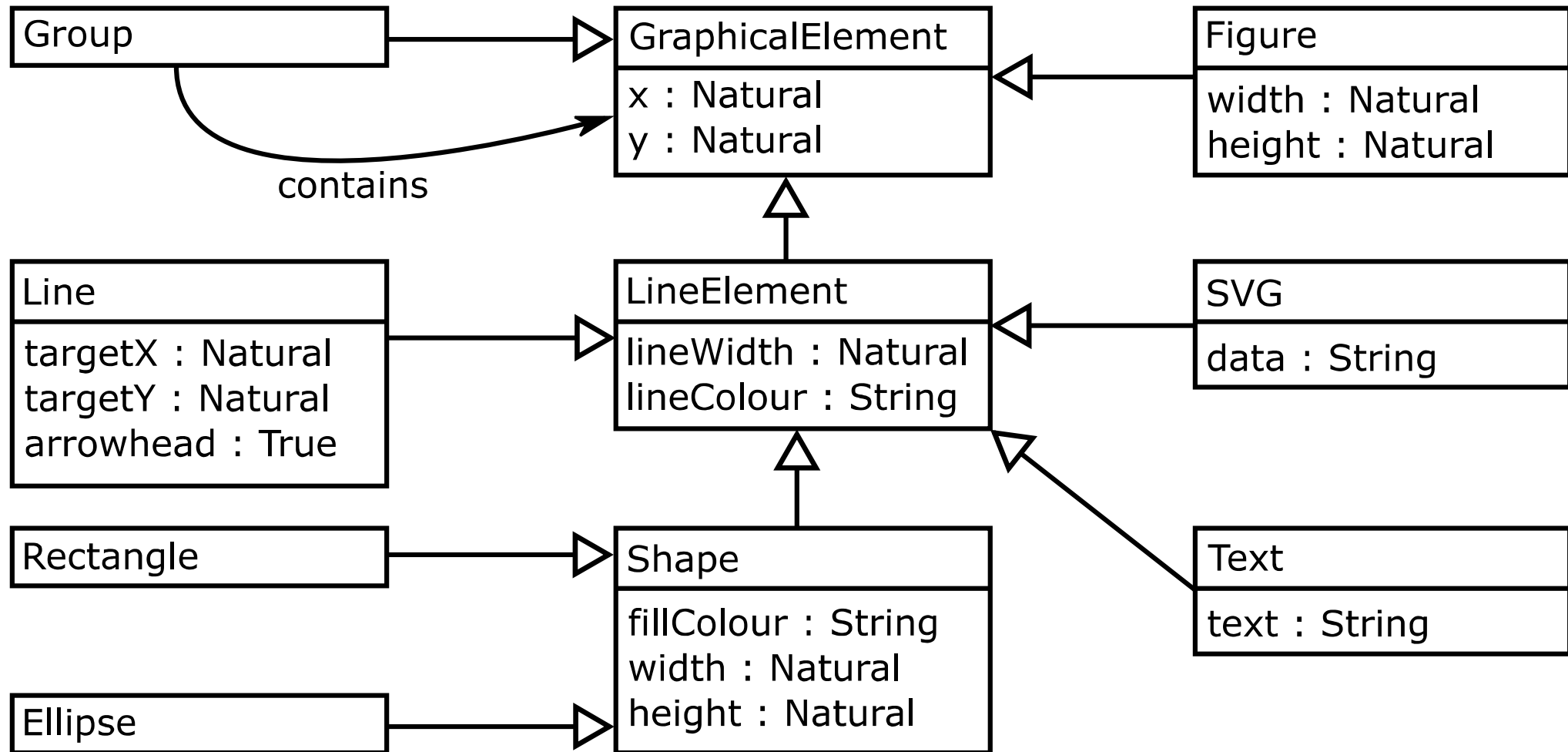
Types of operations



Concrete Syntax



Concrete Syntax:

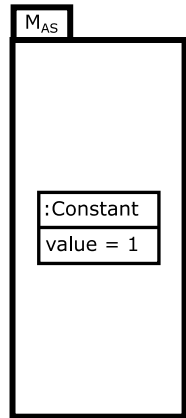


Concrete Syntax (example)

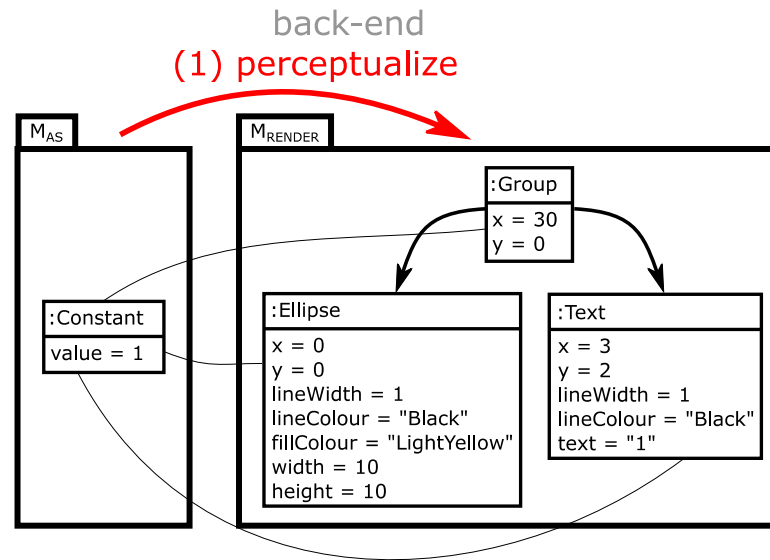
back-end

communication

front-end



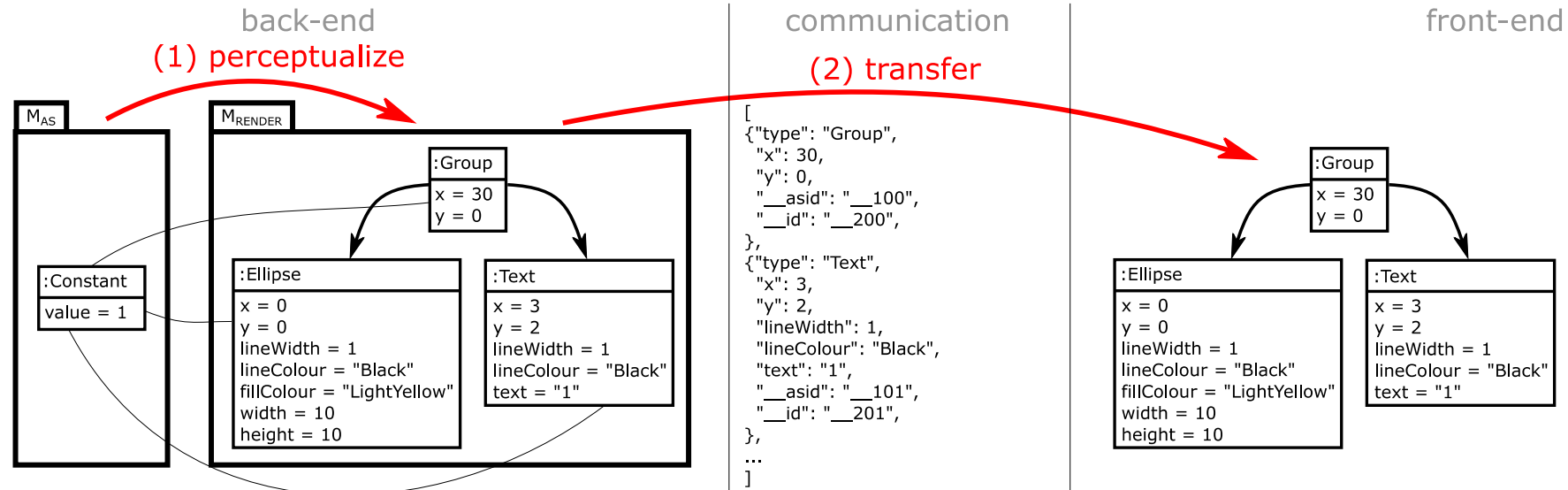
Concrete Syntax (example)



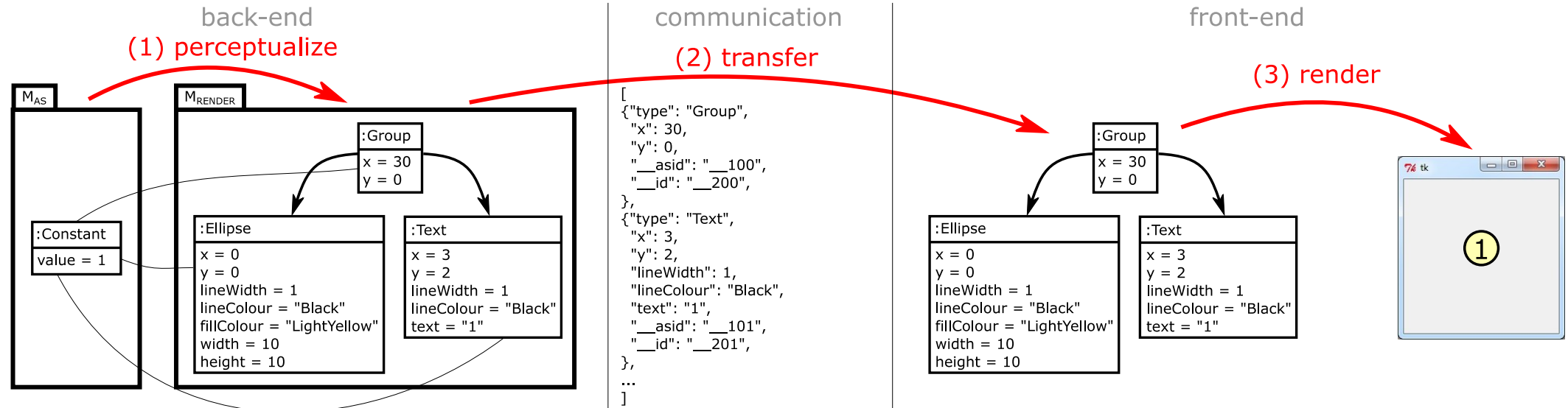
communication

front-end

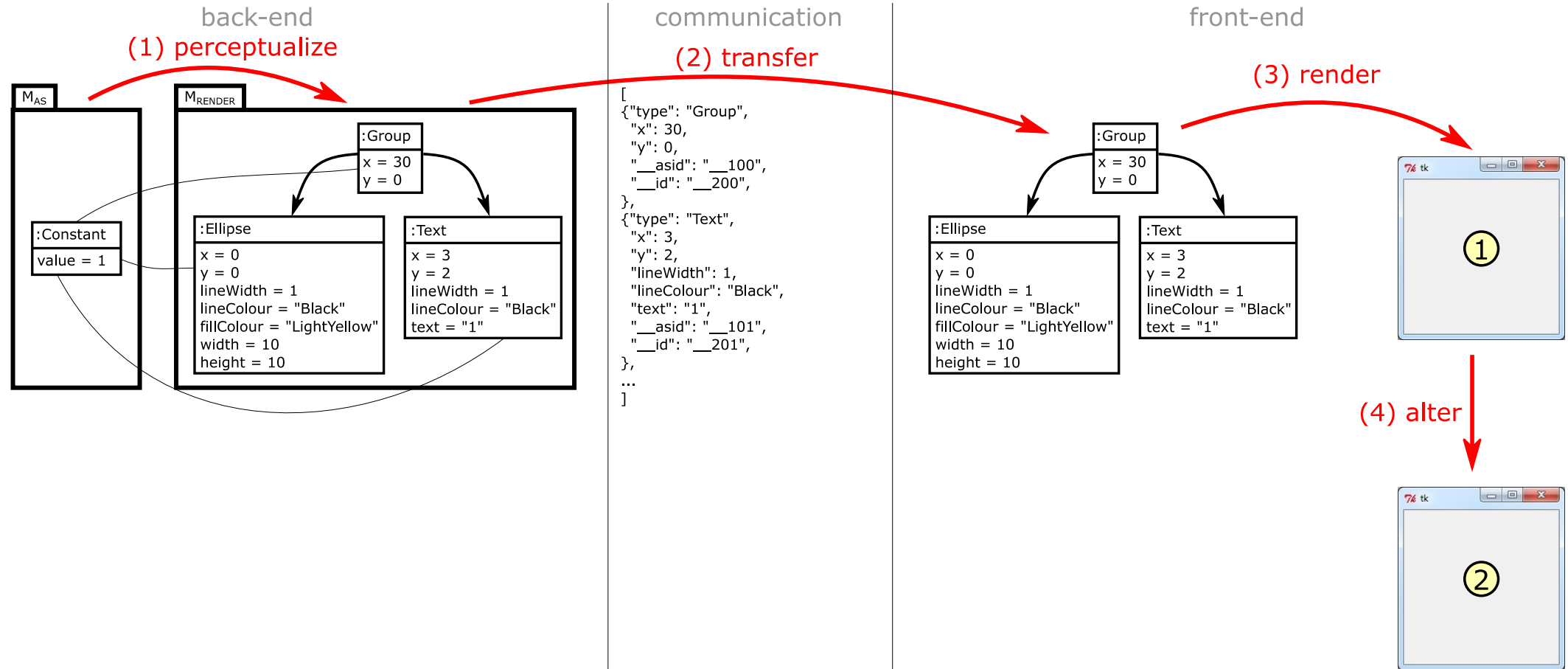
Concrete Syntax (example)



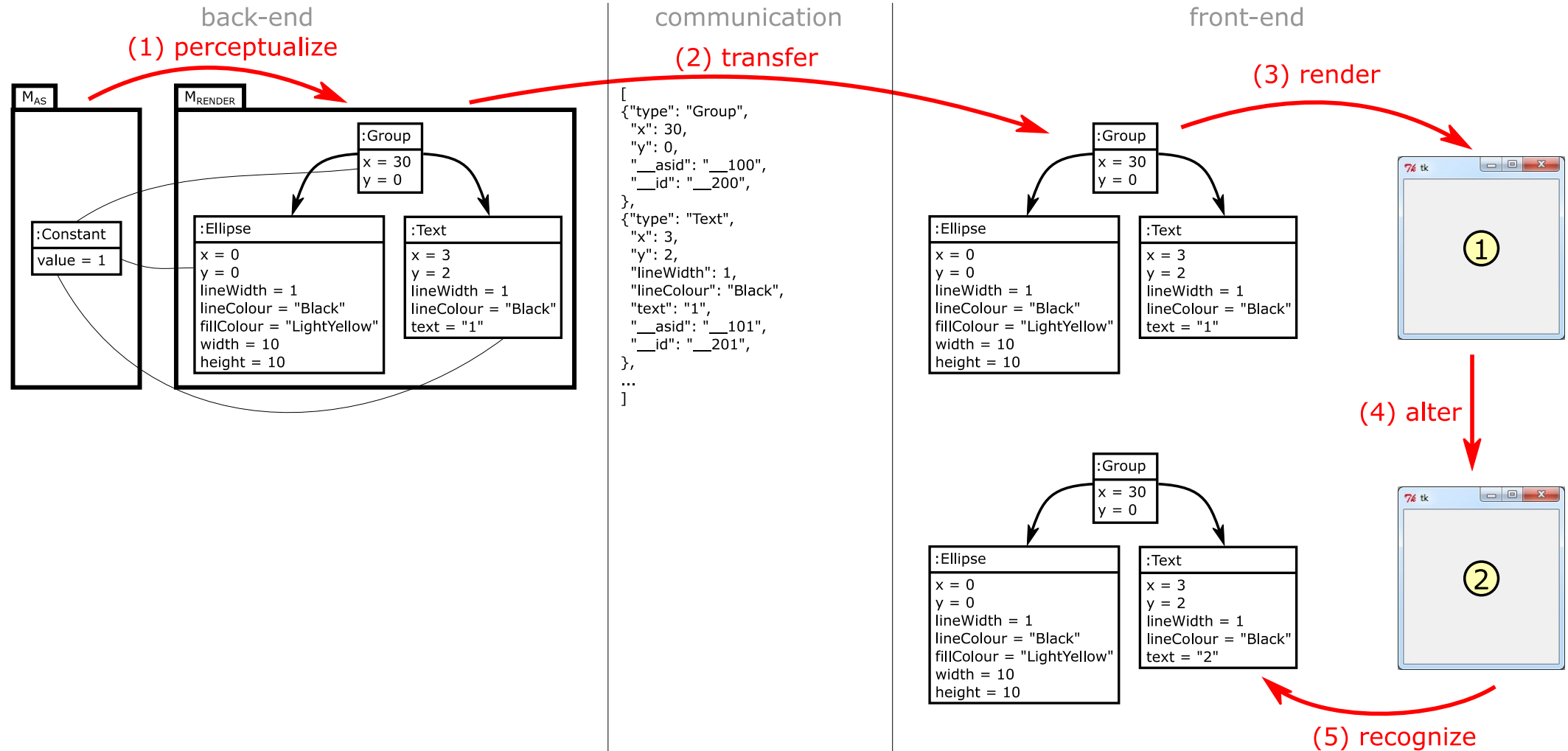
Concrete Syntax (example)



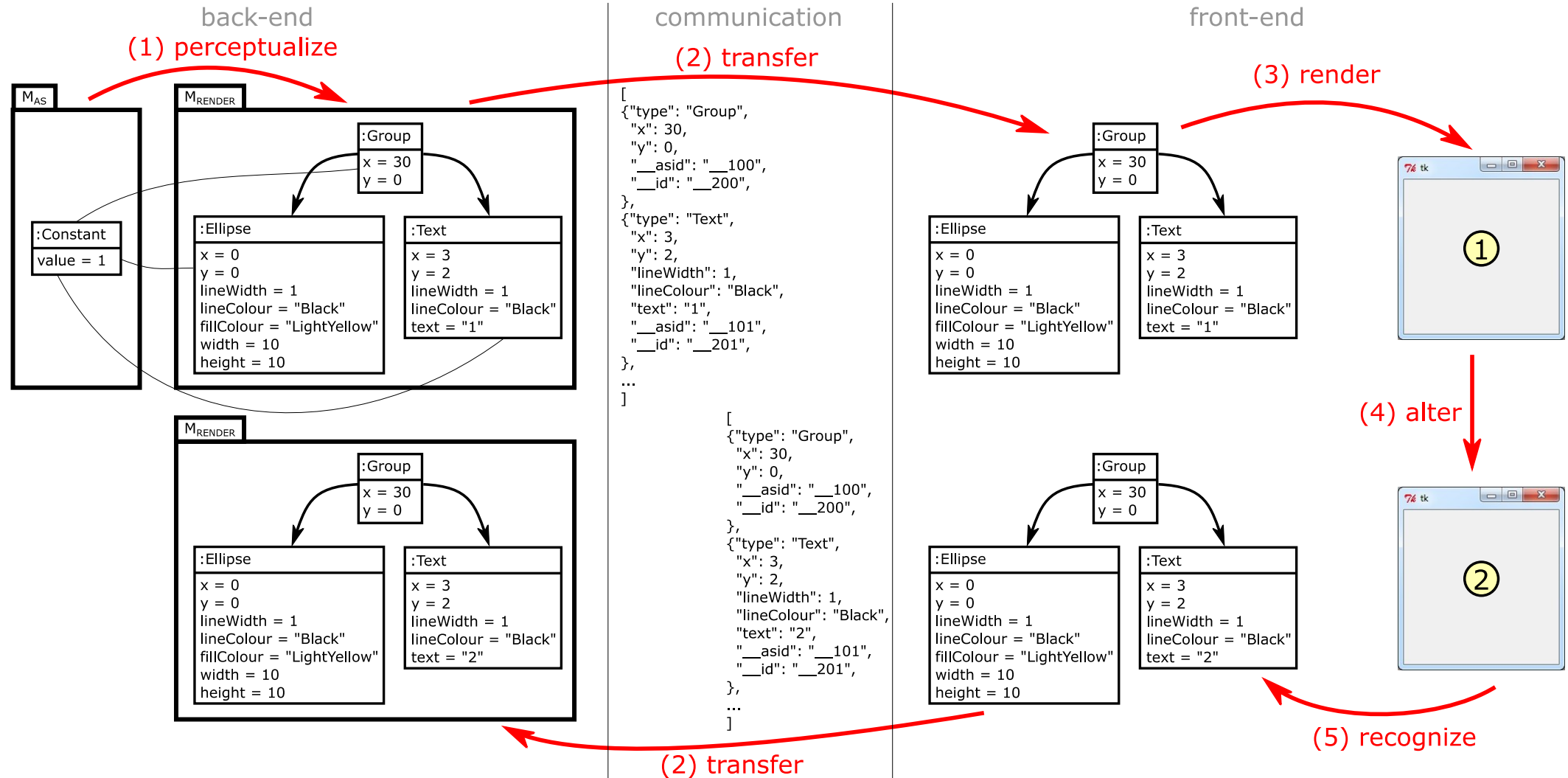
Concrete Syntax (example)



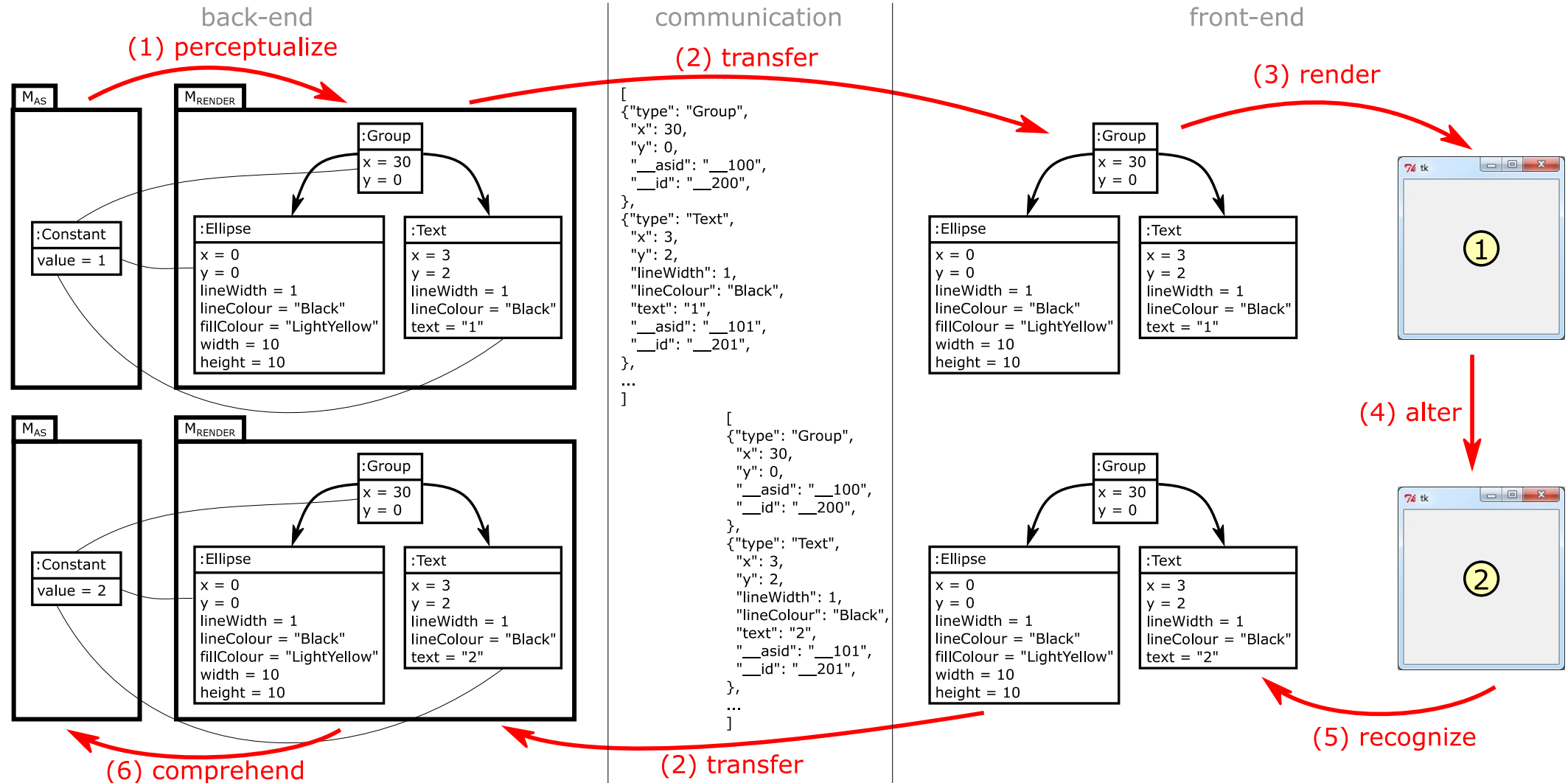
Concrete Syntax (example)



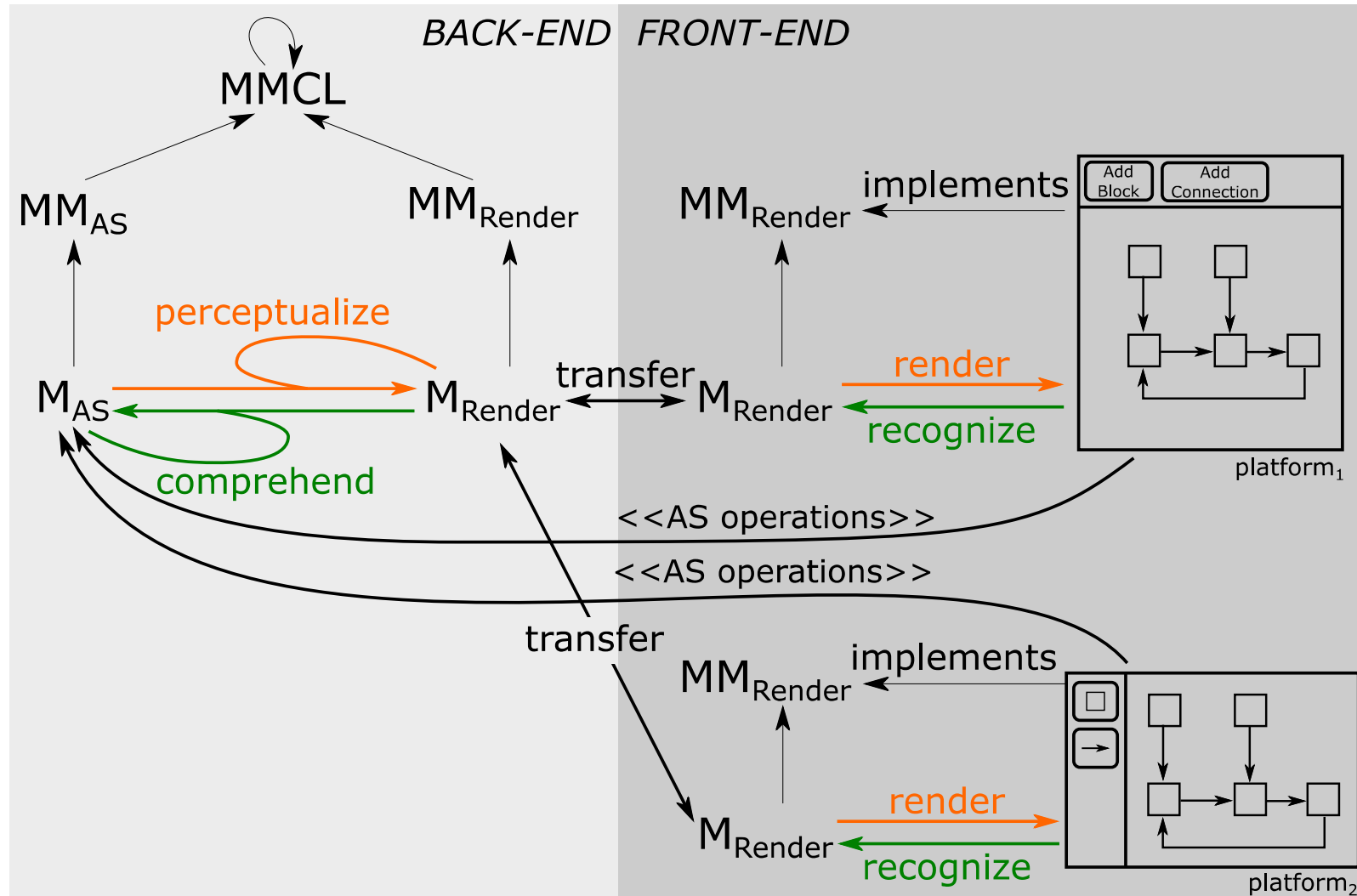
Concrete Syntax (example)



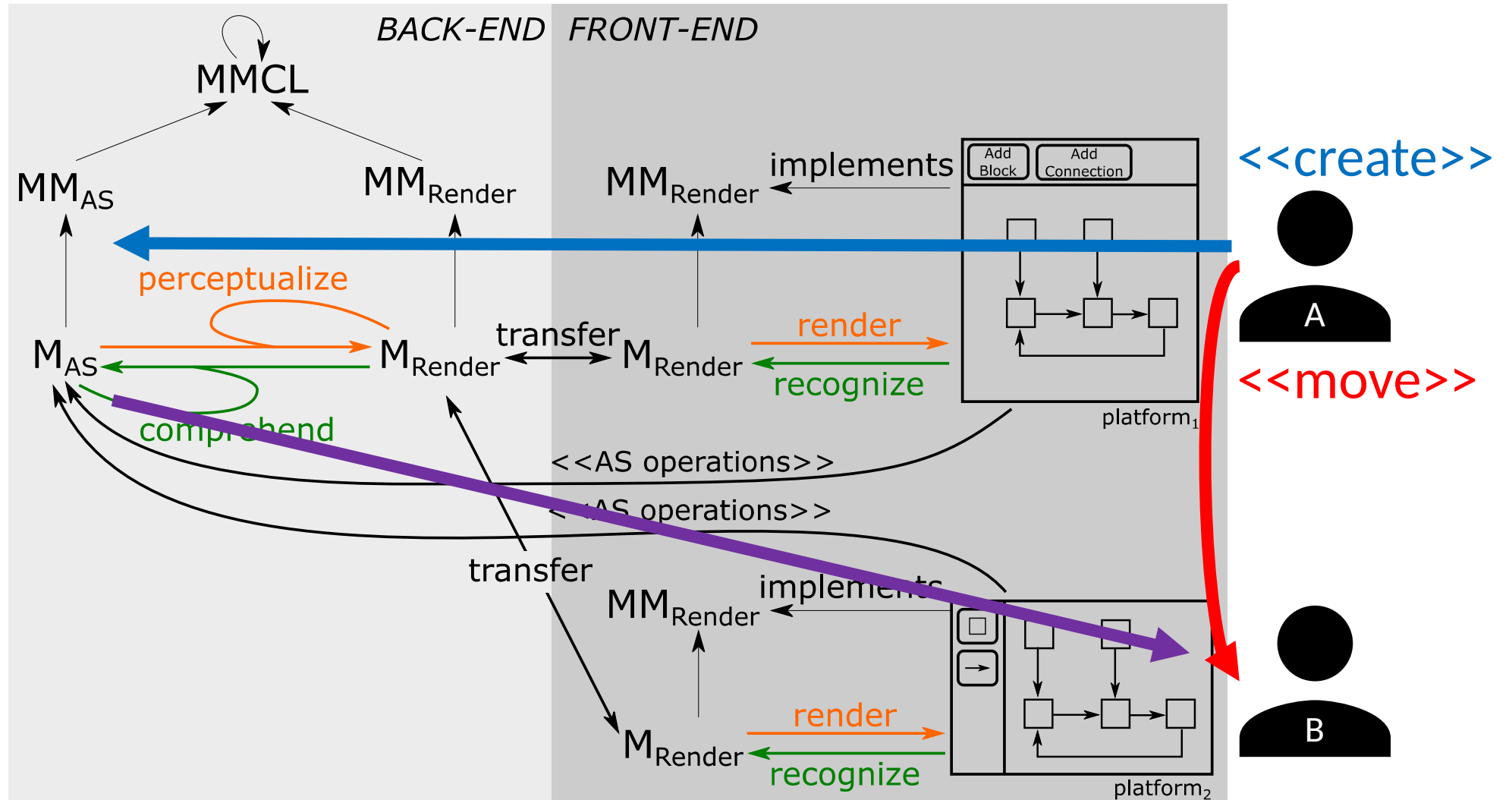
Concrete Syntax (example)



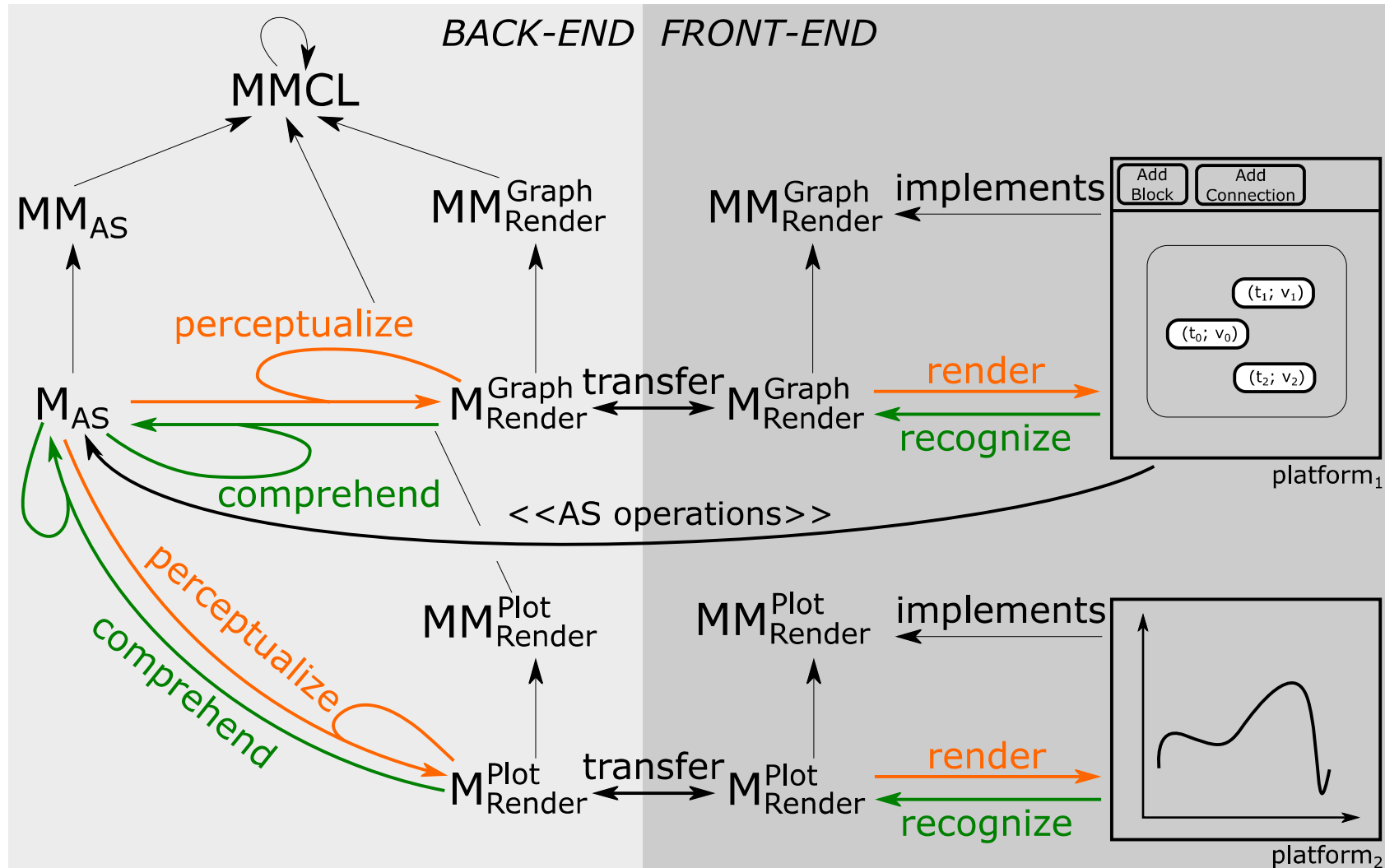
Screenshare



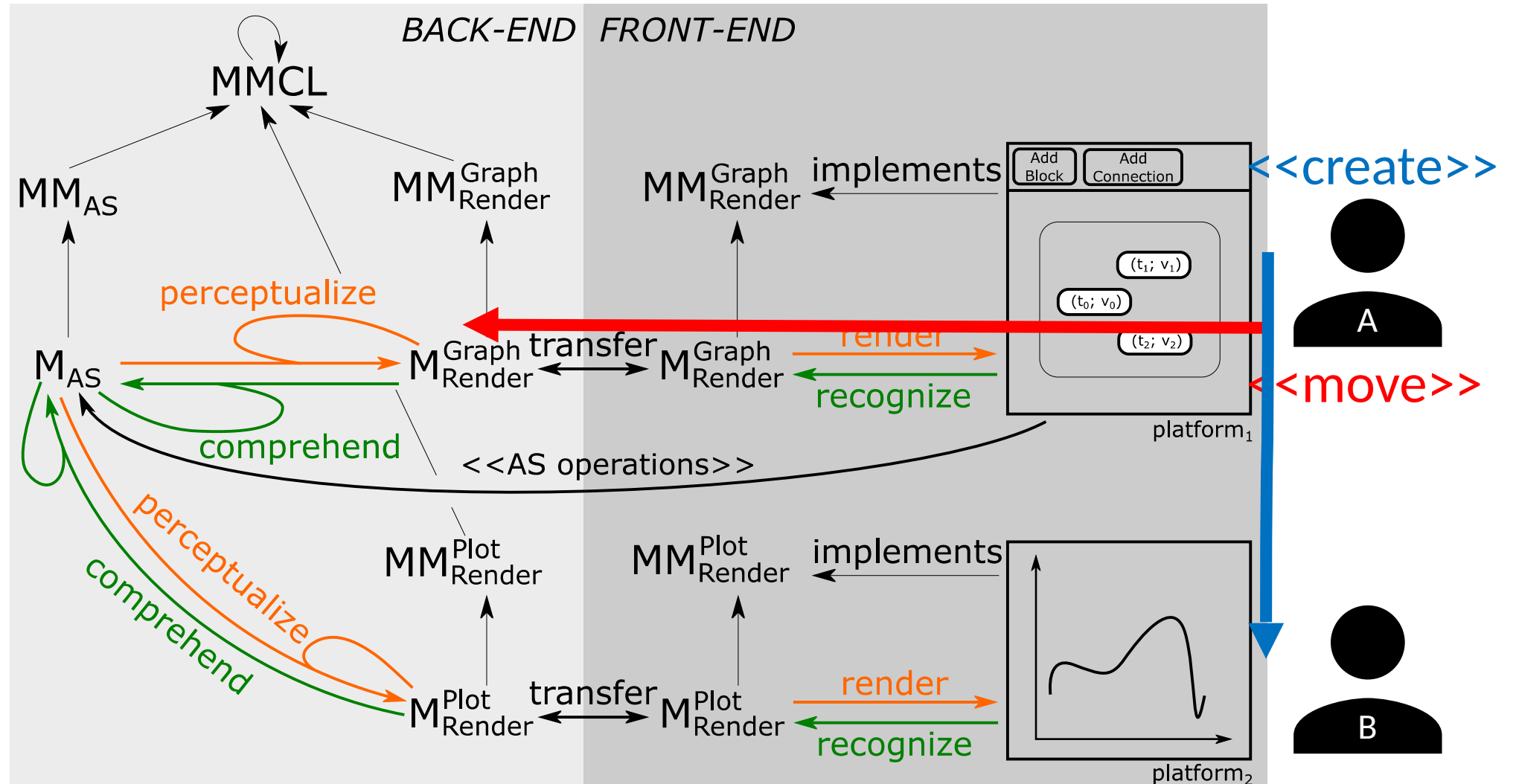
Screenshare (example)



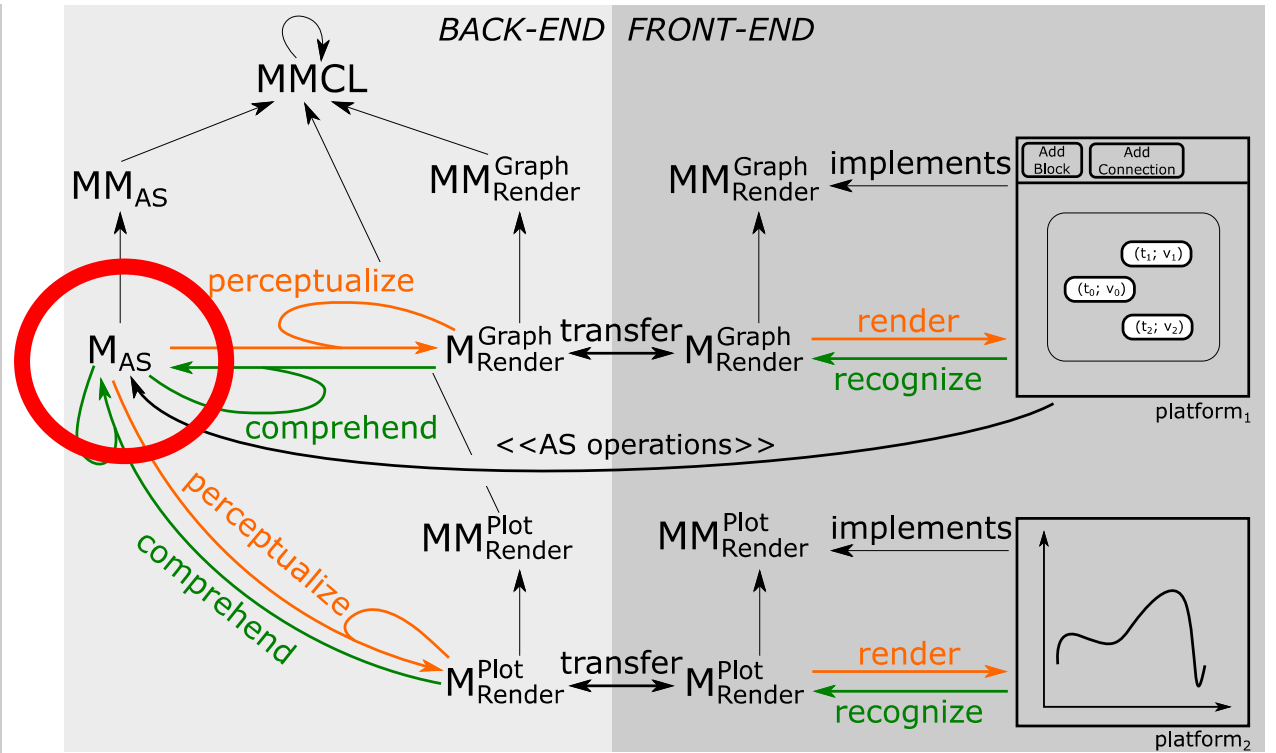
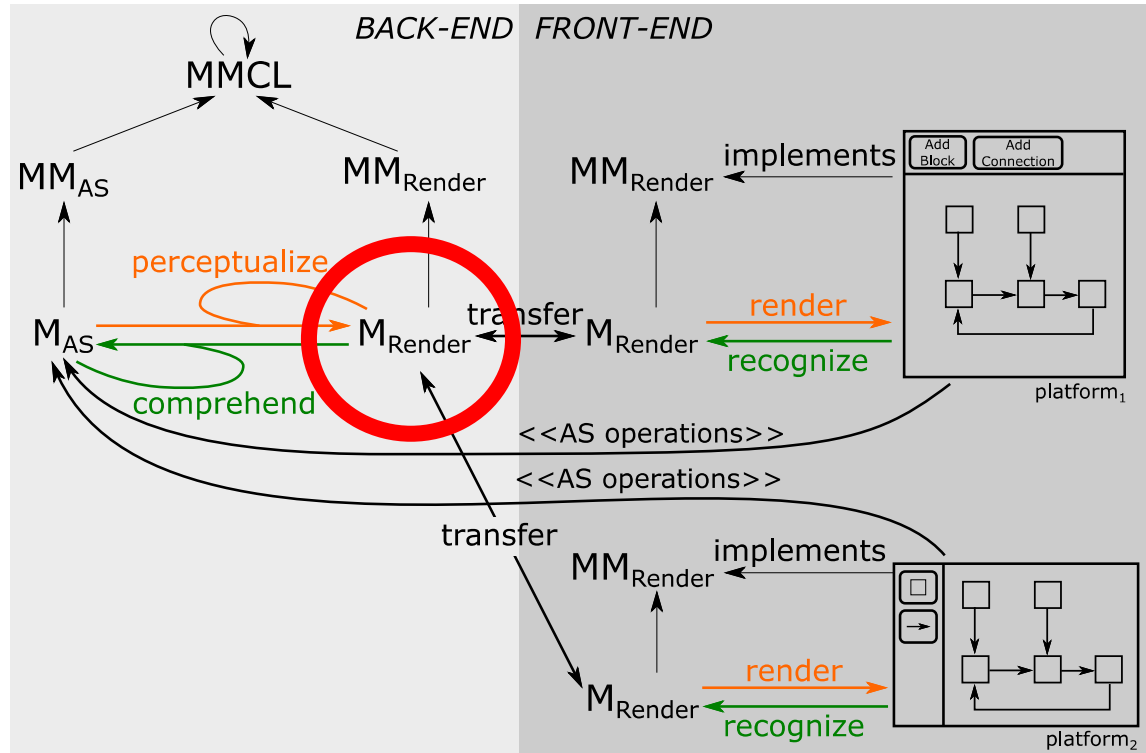
Modelshare



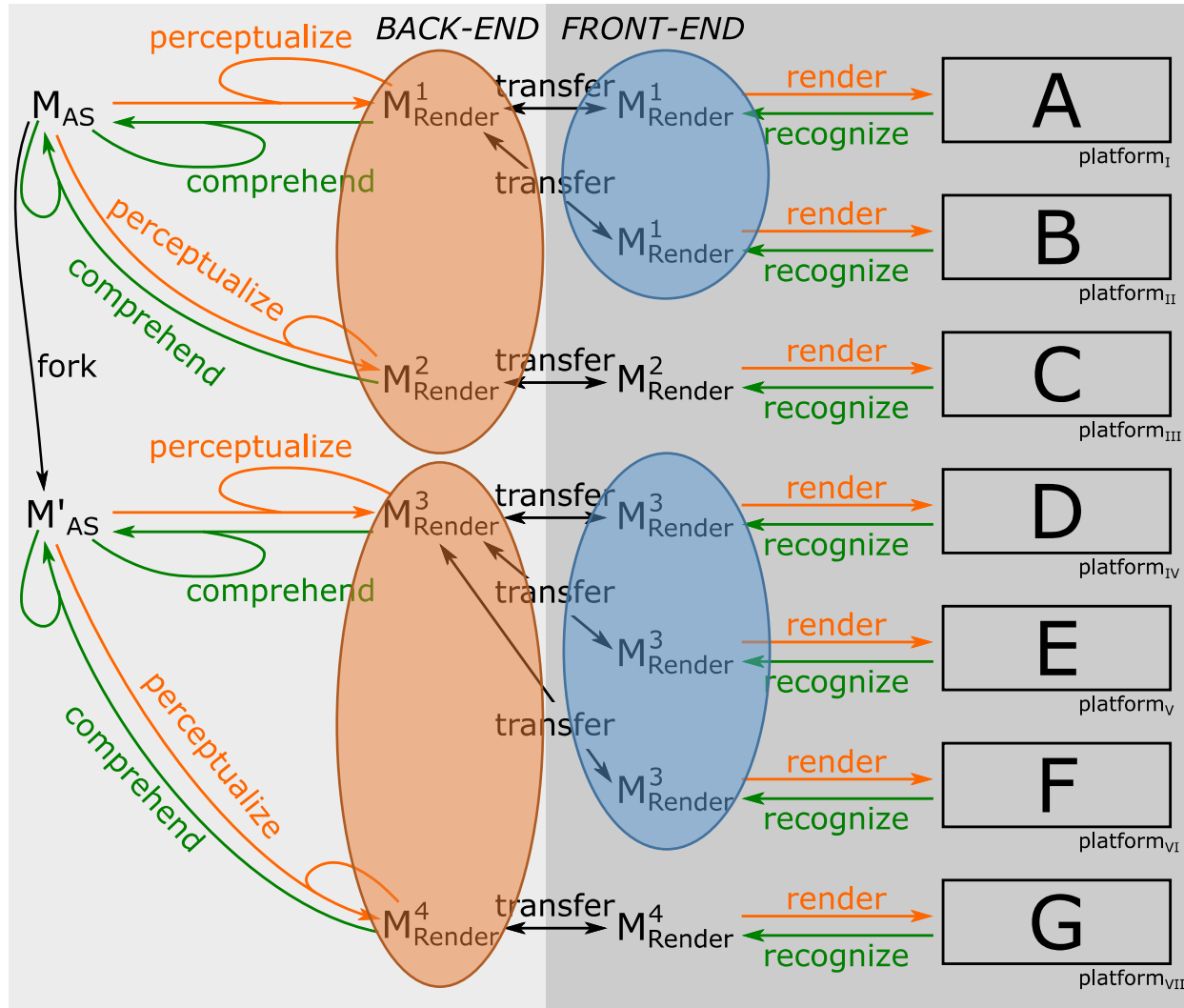
Modelshare (example)



Combination



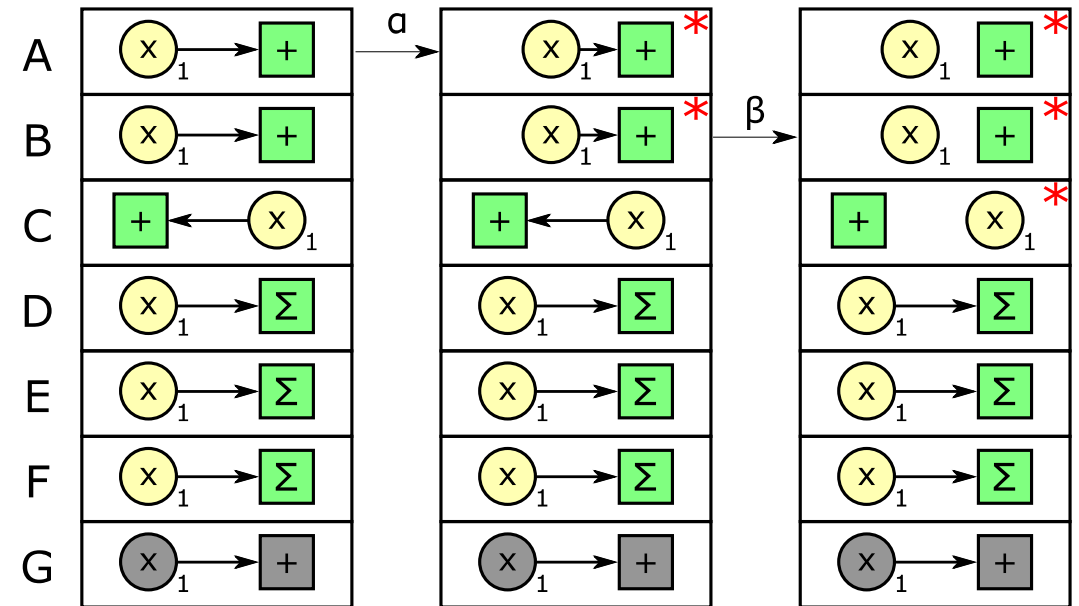
Combination



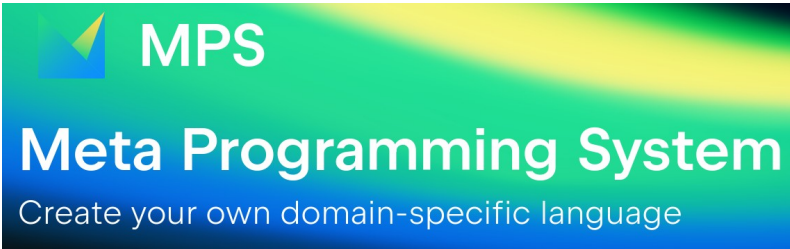
No sharing

Modelshare

Screenshare



projectional editing: only allow Abstract Syntax edit operations



```
System.out.println(String.valueOf(( $\sum \begin{bmatrix} 1 & k & 0 \\ 0 & 1.0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ )));
```

```
System.out.println(exp(a + i * b) - exp(a) * (cos(b) + i * sin(b)));
```

```
matrix<Double> s =  $\begin{bmatrix} \begin{bmatrix} 3.0 \\ 2 \\ 3 \end{bmatrix} & \begin{bmatrix} \sin(1) \\ 1 \\ 7 - \frac{1.0}{2} + 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 3 + \frac{1.0}{2} \\ \exp(1) \end{bmatrix} & \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 4 \end{bmatrix} & \begin{bmatrix} 2 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix};$ 
```