



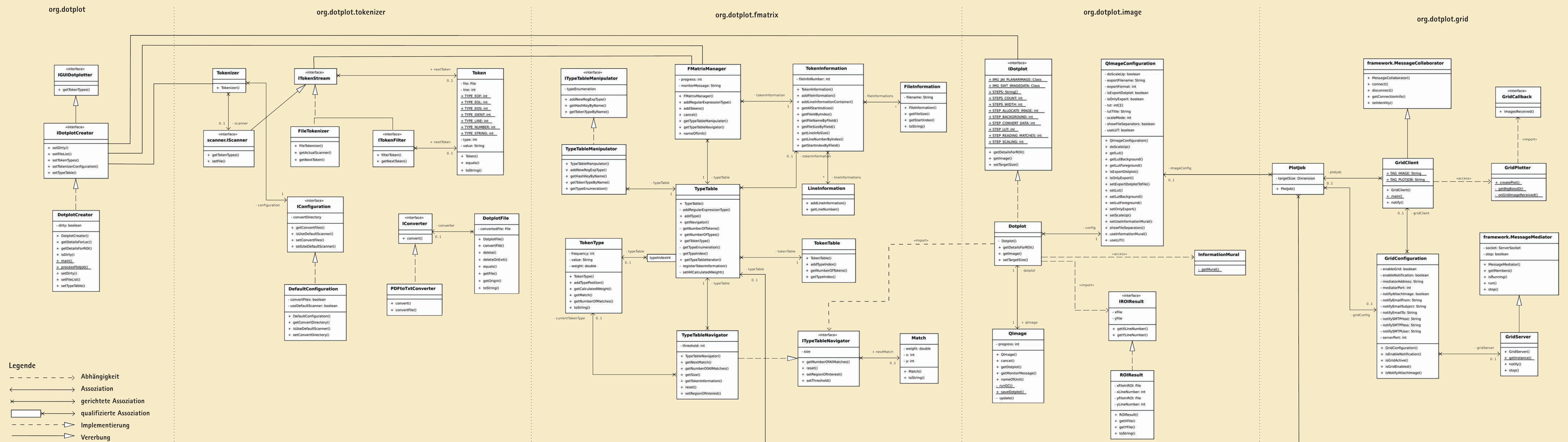
# Master-Entwicklungsprojekt

UNIVERSITY OF APPLIED

## „Visualisieren und Auswerten großer Datenmengen“

## Anwendungsdomäne: Media Systems

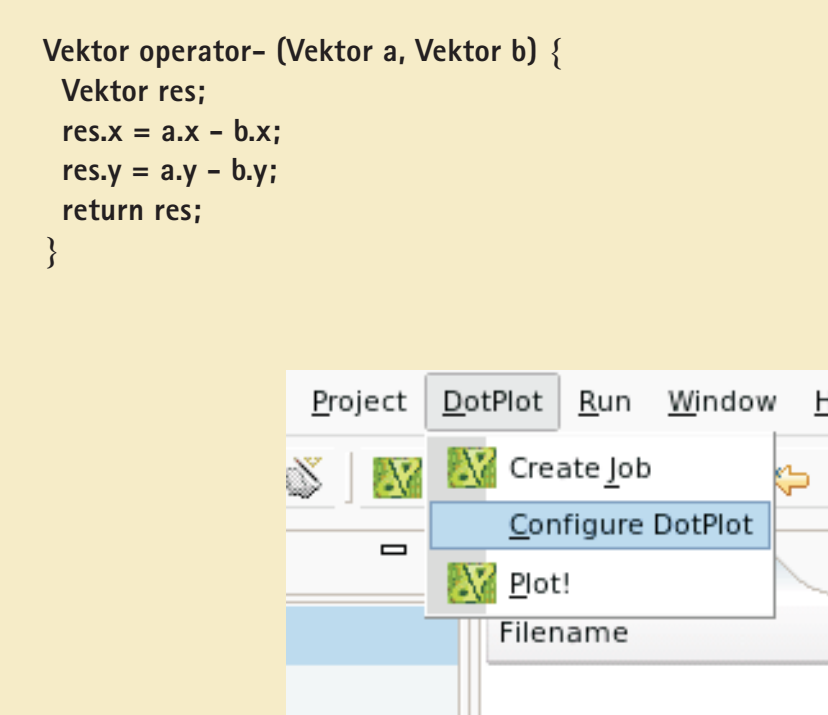
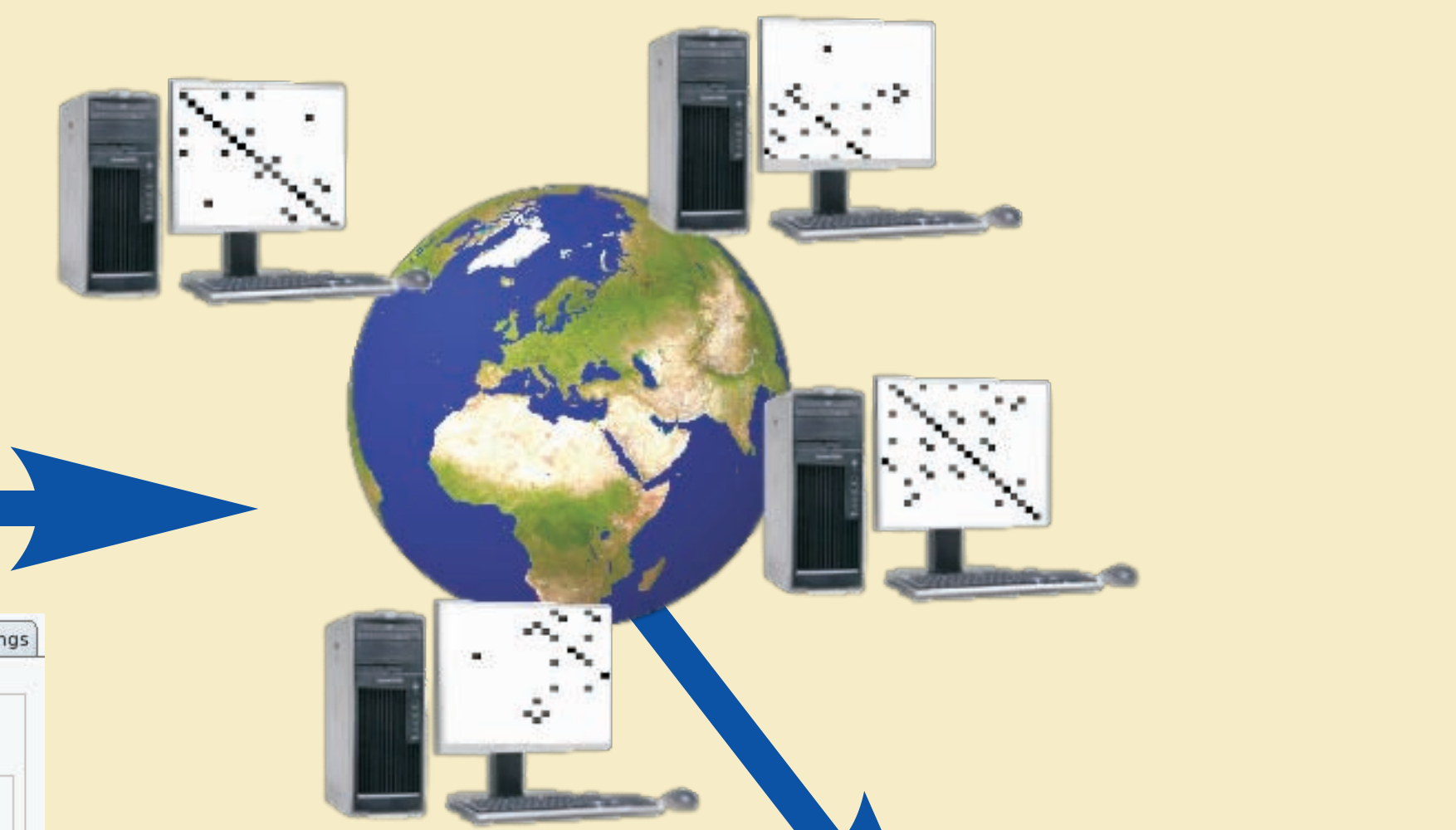
## Zeitstrahl der Dotplot-Erzeugung entlang der Plugin-Architektur



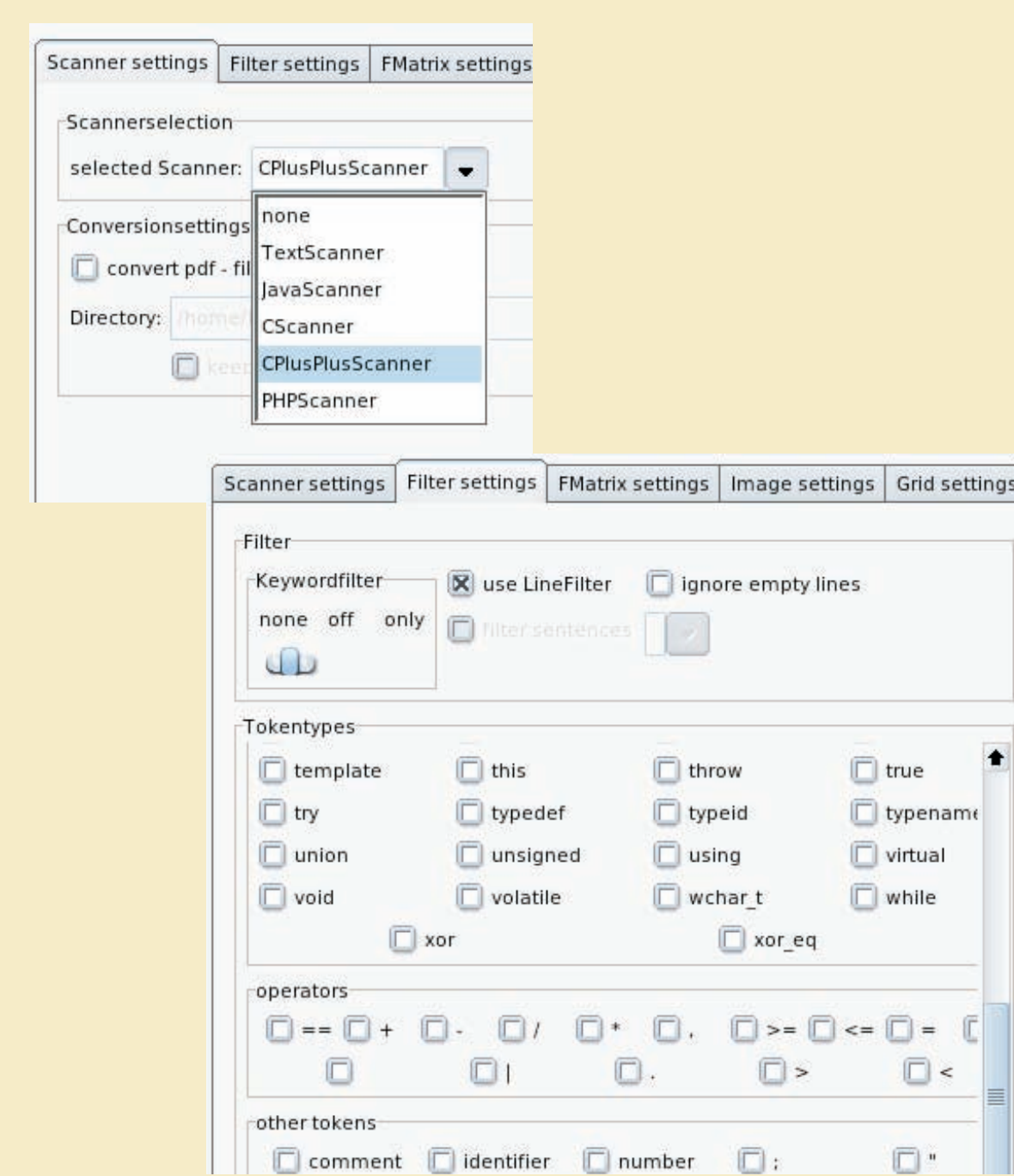
Das Ziel der ersten Phase ist die Erkennung gewisser Grundsymbole, so genannter Tokens, in einem Zeichenstrom aus den Quelldateien. Hierbei können Filter auf die Eingabe angewendet oder die Tokens durch eines ihrer Attribute ersetzt werden. Diese Phase entspricht der lexikalischen Analyse eines Compilers. Die Erkennung von Schlüsselwörtern, Bezeichnern, Operatoren etc. verschiedener Programmiersprachen ermöglicht eine differenziertere Arbeitsweise mit den Plots. Ein PDF-Konverter ermöglicht zusätzlich das Einlesen von PDF-Dateien.

Nach der Umwandlung der Daten in Tokens werden diese zu einer Matrix aus Gleitkommazahlen verarbeitet. Die tatsächliche textuelle Repräsentation der Tokens spielt im Weiteren keine Rolle, weshalb nur noch Tokenarten unterschieden werden. So enthält der Satz "Wenn Fliegen hinter Fliegen fliegen fliegen Fliegen hinter Fliegen" zwar neun Tokens (auf das Komma wurde verzichtet), aber nur vier verschiedene (ließe man die Großschreibung außer Acht, wären es sogar nur drei). Da häufige Treffer einen geringeren Informationsgehalt als seltene haben, ist es sinnvoll, sie zu gewichten. Das ermöglicht, den Rechenaufwand erheblich zu verringern, indem man Tokens, deren Häufigkeit einen bestimmten Grenzwert überschreitet, nicht weiter berücksichtigt.

Im letzten Schritt muss die speicherinterne Darstellung der Matrix in ein Bild umgewandelt werden. Der Rechenaufwand ist hoch, da hier jedes einzelne Pixel eines unter Umständen extrem großen Bildes berechnet werden muss. Da der Plot in aller Regel nicht auf den Bildschirm passt, muss die Darstellung interpoliert werden. Dazu verwendet das Plugin den Information-Mural-Algorithmus. An dieser Stelle erlaubt das Plugin die Verteilung des Rechenaufwandes auf mehrere Rechner in einem Grid.



```
Token "Vektor" in line 1
Token "operator" in line 1
Token "Vektor" in line 1
Token "a" in line 1
Token "-" in line 1
Token "Vektor" in line 1
Token "b]" in line 1
Token "]" in line 1
EOLToken
Token "Vektor" in line 2
Token "res" in line 2
Token "-" in line 2
EOLToken
Token "res" in line 3
Token "-" in line 3
Token "x" in line 3
Token "-" in line 3
Token "a" in line 3
Token "-" in line 3
Token "x" in line 3
Token "-" in line 3
Token "b" in line 3
Token "-" in line 3
Token "x" in line 3
Token "-" in line 3
EOLToken
Token "res" in line 4
Token "-" in line 4
Token "y" in line 4
Token "-" in line 4
Token "a" in line 4
Token "-" in line 4
Token "y" in line 4
```



Matches: 124  
Tokens: 41  
Types: 18

X: 39 Y: 39 W:

X: 39.7; 39 Weight: 1.0  
X: 38.7; 38 Weight: 1.0  
X: 35.7; 35 Weight: 1.0  
X: 25.7; 25 Weight: 0.3333333333333333  
X: 24.7; 24 Weight: 0.3333333333333333  
X: 33.7; 25 Weight: 0.3333333333333333  
X: 25.7; 29 Weight: 0.3333333333333333  
X: 29.7; 29 Weight: 0.3333333333333333  
X: 33.7; 29 Weight: 0.3333333333333333  
X: 38.7; 29 Weight: 0.3333333333333333  
X: 29.7; 33 Weight: 0.3333333333333333  
X: 33.7; 33 Weight: 0.3333333333333333  
X: 19.7; 19 Weight: 0.5  
X: 31.7; 19 Weight: 0.5  
X: 18.7; 18 Weight: 0.5  
X: 30.7; 18 Weight: 0.5  
X: 18.7; 30 Weight: 0.5  
X: 31.7; 31 Weight: 0.5  
X: 18.7; 18 Weight: 0.5  
X: 18.7; 30 Weight: 0.5  
X: 14.7; 14 Weight: 0.5  
X: 26.7; 14 Weight: 0.5  
X: 14.7; 26 Weight: 0.5  
X: 26.7; 26 Weight: 0.5  
X: 17.7; 17 Weight: 0.3333333333333333  
X: 17.7; 13 Weight: 0.3333333333333333  
X: 17.7; 13 Weight: 0.3333333333333333  
X: 21.7; 13 Weight: 0.3333333333333333  
X: 13.7; 17 Weight: 0.3333333333333333  
X: 17.7; 17 Weight: 0.3333333333333333  
X: 17.7; 17 Weight: 0.3333333333333333

X: 32.25	28 Weight: 0.16666666666666666	)
X: 12.75	32 Weight: 0.16666666666666666	)
X: 16.75	32 Weight: 0.16666666666666666	)
X: 20.75	32 Weight: 0.16666666666666666	)
X: 24.75	32 Weight: 0.16666666666666666	)
X: 28.75	32 Weight: 0.16666666666666666	)
X: 32.75	32 Weight: 0.16666666666666666	)
X: 10.75	10 Weight: 0.25	ES050
X: 22.75	10 Weight: 0.25	Vector
X: 34.75	10 Weight: 0.25	alpha
X: 37.75	10 Weight: 0.25	oper
X: 10.75	22 Weight: 0.25	bits
X: 22.75	22 Weight: 0.25	return
X: 34.75	22 Weight: 0.25	res
X: 37.75	22 Weight: 0.25	x
X: 10.75	34 Weight: 0.25	y
X: 22.75	34 Weight: 0.25	z
X: 34.75	34 Weight: 0.25	{
X: 37.75	34 Weight: 0.25	}
X: 22.75	37 Weight: 0.25	0
X: 34.75	37 Weight: 0.25	1
X: 37.75	37 Weight: 0.25	2
X: 9.75	9 Weight: 0.25	
X: 19.75	9 Weight: 0.25	
X: 23.75	9 Weight: 0.25	
X: 36.75	9 Weight: 0.25	
X: 9.75	11 Weight: 0.25	
X: 11.75	11 Weight: 0.25	

Frequency:

4

Regular Expression

Add Type

ApplyReset

