



# VU Research Portal

## Gatekeeping in the Digital Age

Welbers, K.

2016

### **document version**

Publisher's PDF, also known as Version of record

[Link to publication in VU Research Portal](#)

### **citation for published version (APA)**

Welbers, K. (2016). *Gatekeeping in the Digital Age*. Vrije Universiteit.

### **General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

### **Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

### **E-mail address:**

[vuresearchportal.ub@vu.nl](mailto:vuresearchportal.ub@vu.nl)

```
library(Matrix)
library(igraph)
library(png)
library(scales)
```

```
## save text as image
png(filename="tmp/title.png")
plot.new(); text(0.5,0.5, "Gatekeeping in the Digital Age", cex = 5)
dev.off()
```

```
## read image as matrix and extract coordinates (silly hack, but works)
img = readPNG("tmp/title.png")
m = img[, 1]
m = t(m[,nrow(m), 1])
m = Matrix(m, sparse=T)
co = which(m == 1, arr.ind = T)
```

```
## group coordinates according to words
word1 = "Gatekeeping"
word2 = "in"
word3 = "the"
word4 = "Digital"
word5 = "Age"

word1[1] <- 240 & col1 = 320
word1[2] <- 240 & col1 = 320
word1[3] <- 240 & col1 = 320
word1[4] <- 250 & col1 = 750
word1[5] <- 175 & col1 = 750

## create nodes
ord = order(m)
co = co[ord,]
word = word[ord]
```

```
## take a random sample of coordinates per word to manage node density
fit1 = sample(which(m == 1), size = 200)
fit = c(fit1, sample(which(m == 1), size = 200, replace=T))
fit = c(fit, sample(which(m == 1), size = 200, replace=T))
fit = c(fit, sample(which(m == 1), size = 200, replace=T))
fit = fit[order(fit)]
word = word[fit]
co = co[fit,]
```

```
## create edges between pairs sampled from the igraph network
edges = data.frame(x=2, nrow=(5+1)*2, y=2, row=(co+1))
edges[word == 1, 1] = word[ord]
edges[word %in% 2:3, 1] = sample(1:200, size = word == 1, 2000)
edges[word == 4, 1] = edges[word %in% 2:3, 1]
edges[word == 5, 1] = edges[word == 4, 1]
edges = rbind(edges, data.frame(x=2, nrow=(5+1)*2, y=2, row=(co+1))
```

```
m = sparseMatrix(nrow=co+2, ncol=co+2, x=edges[,2], rep(1, nrow(edges)))
g = graph.adjacency(m)
```

```
## move words around, e.g. to the right
col[word==2, 1] = col[word==2, 1] + 20
col[word==3, 1] = col[word==3, 1] + 20
col[word==3, 2] = col[word==3, 2] - 30
col[word==4, 1] = col[word==4, 1] + 20
col[word %in% 4:5, 2] = col[word %in% 4:5, 2] + 20
col[word %in% 5, 1] = col[word %in% 5, 1] + 20
```

```
lay = rbind(t(matrix(c(0,1,6), nrow=2)),
            layout.norm(co),
            t(matrix(c(-1,6), nrow=2)))
```

```
## set network attributes for visualization
pal = terrain.colors
V(g)$color = pal[9][c(6, word, 7)]
V(g)$frame.color = 'white'
V(g)$size = 1
V(g)$size[word %in% c(2,3,5)] = 0.7
V(g)$label = sample(0:1, replace=T, vcount(g)) ## add 0/1 as node labels..
V(g)$label.cex = 0.16 ## for fancy digital feel
V(g)$label.color = alpha("black", 0.6) ## though far too small for anyone to notice
```

```
V(g)$size[c(1, vcount(g))] = 8
V(g)$label[c(1, vcount(g))] = paste(sapply(split(sample(0:1, replace=T, 86), rep(1:7, c(6,12,6,6,6,6,6))), paste("collapse=", collapse="r"))
V(g)$color[1] = pal[6][1]
V(g)$color[vcount(g)] = pal[6][6]
V(g)$frame.color[c(1, vcount(g))] = pal[6][5]
V(g)$frame.width=3
```

```
g = delete.edges(g, sample(1:ecount(g), ecount(g)-200))
E(g)$width = mnorm(ecount(g), 4)
E(g)$width[sample(1:ecount(g), ecount(g)-60)] = mnorm(ecount(g)-60,0.0001)
E(g)$curved = rep(seq(0.3,-0.3,length.out=ecount(g)/2), 2)
E(g)$color = pal[9, alpha=0.6][c(6, word, 7)][get.edges(g, E(g))[,2]]
E(g)$arrow.size = 0.00001 # go away arrow, nobody likes you
```

```
## ignore top node (then why include it in the first place?)
g = delete.vertices(g, 1)
lay = lay[-1,]
```

```
## read this code as text to plot as background (Droste, eat your heart out)
thiscode = readLines("create_coverpage.r")
thiscode = paste(thiscode, collapse="\n")
```

```
## TADA!!!
pdf("coverpage.pdf", width = 17/2.54, height = 24/2.54)
par(mar=c(0,0,0,0), bg="black")
plot.new()
text(0.0,0, thiscode, col=alpha("green", 0.3), cex=0.35, font=0)
text(-1.4, -1.4, "Gatekeeping in the Digital Age", col="white", cex=5, font=0)
par(mar=c(-3,-3,-3,-3))
plot(g, layout=lay)
text(-1.015,-1.4, "> Kasper Welbers", col=alpha("white", 0.8), cex=1.7, adj=0)
dev.off()
```

> Kasper Welbers