BACK TO THE FUTURE?

~1970


Bell Laboratories

AT&T Archives: The UNIX Operating System (http://www.youtube.com/watch?v=tc4ROCJYbm0)

2013

UNIX®

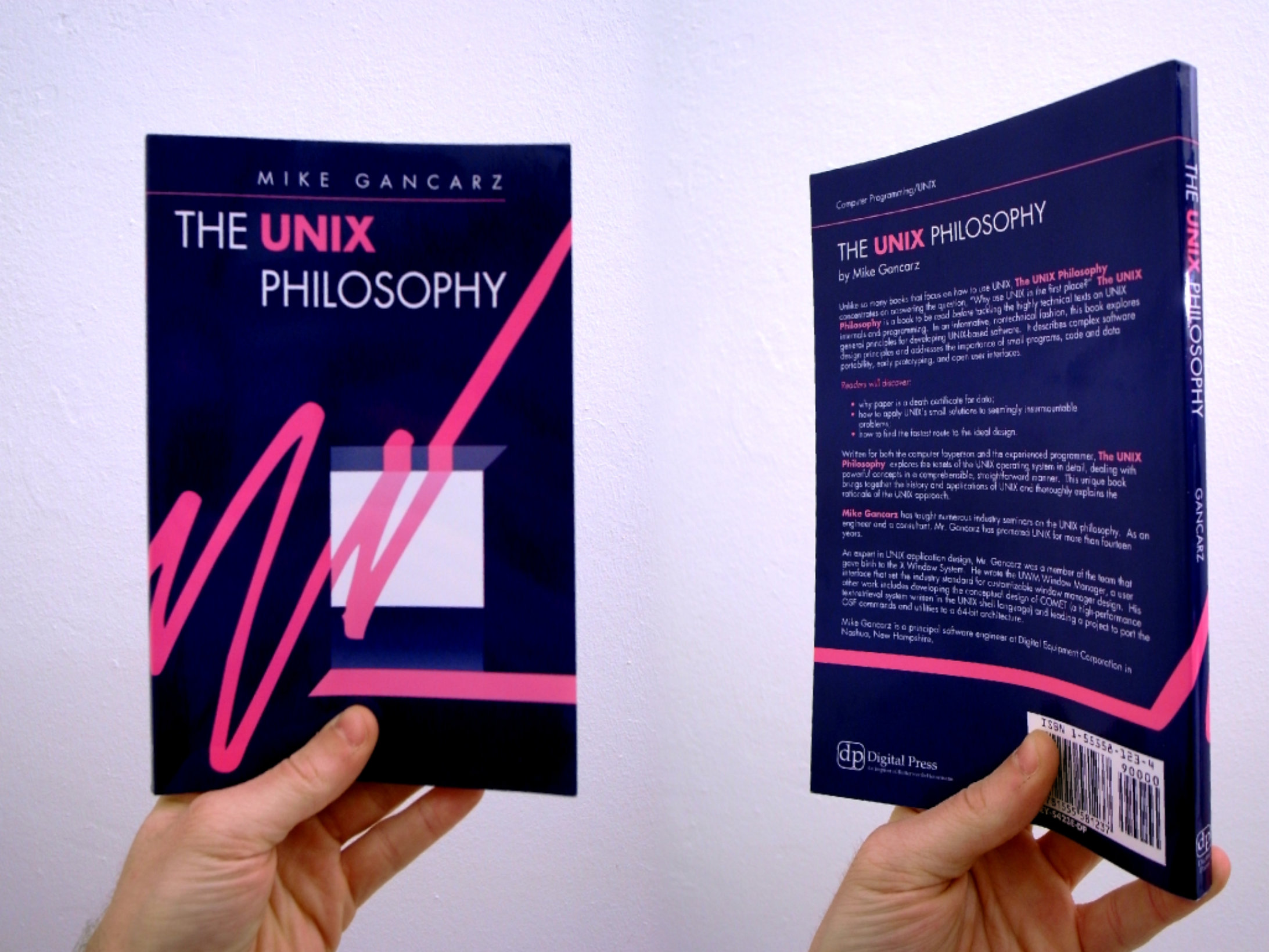
Celebrating 40 years uptime

http://www.unix.org/

1983

GNU's Not Unix!

2007


MIKE GANCARZ
THE UNIX PHILOSOPHY
Computer Programming/UNIX
THE UNIX PHILOSOPHY
by Mike Gancarz
Unlike so many books that focus on how to use UNIX, The UNIX Philosophy concentrates on answering the question, "Why use UNIX in the first place?" The UNIX Philosophy is a book to be read before tackling the highly technical texts on UNIX internals and programming. In an informative, nontechnical fashion, this book explores general principles for developing UNIX-based software. It describes complex software design principles and addresses the importance of small programs, code and data portability, early prototyping, and open user interfaces.
Readers will discover:
why paper is a death certificate for data;
how to apply UNIX's small solutions to seemingly insurmountable problems;
how to find the fastest route to the ideal design.
Written for both the computer layperson and the experienced programmer, The UNIX Philosophy explores the tenets of the UNIX operating system in detail, dealing with powerful concepts in a comprehensible, straightforward manner. This unique book brings together the history and applications of UNIX and thoroughly explains the rationale of the UNIX approach.
Mike Gancarz has taught numerous industry seminars on the UNIX philosophy. As an engineer and a consultant, Mr. Gancarz has promoted UNIX for more than fourteen years.
An expert in UNIX application design, Mr. Gancarz was a member of the team that gave birth to the X Window System. He wrote the UWM Window Manager, a user interface that set the industry standard for customizable window manager design. His other work includes developing the conceptual design of COMET (a high-performance text-retrieval system written in the UNIX shell language) and leading a project to port the OSF commands and utilities to a 64-bit architecture.
Mike Gancarz is a principal software engineer at Digital Equipment Corporation in Nashua, New Hampshire.
dp Digital Press
ISBN 1-55558-123-4
90000
THE UNIX PHILOSOPHY
GANCARZ

What?

Small is beautiful.
Make each program do one thing well.
Build a prototype as soon as possible.
Choose portability over efficiency.
Store numerical data in flat ASCII files.
Use software leverage to your advantage.
Use shell scripts to increase leverage and portability.
Avoid captive user interfaces.
Make every program a filter.

Small is beautiful.

Small programs are easy to understand.

Small programs are easy to maintain.

Small programs consume fewer system resources.

Small programs are easier to combine with other tools.

Make each program do one thing well.

The best program does no more but
one task in its life and does it well.

The program is loaded into memory,
accomplishes its function,
and then gets out of the way to allow
the next single-minded program to begin.

Build a prototype as soon as possible.

Prototyping is a learning process.

Early prototyping reduces risk.

Choose portability over efficiency.

Next ...'s hardware will run faster.

Don't spend too much time
making a program run faster.

The most efficient way is rarely portable.

Good programs never die – they are ported
to new hardware platforms.

Store numerical data in flat ASCII files.

ASCII text is a common interchange format.

ASCII text is easily read and edited.

ASCII data files simplify the use of Unix text tools.

Increased portability overcomes the lack of speed
(of flat ASCII text files...)

The lack of speed is overcome by next year's machine.

Use software leverage to your advantage.

Good programmers write good code;
great programmers "borrow" good code.

Avoid the not-invented-here syndrome.

Allow other people to use your code
to leverage their own work.

Automate everything.

Use shell scripts to increase leverage and portability.

Shell scripts give you awesome leverage.

Shell scripts leverage your time, too.

Shell scripts are more portable than C.

Resist the desire to rewrite shell scripts in C.

Avoid captive user interfaces.

*CUI*s assume that the user is human.

CUI command parsers are often big and ugly to write.

*CUI*s tend to adopt a "big is beautiful" approach.

Programs having *CUI*s are hard to combine with other programs.

*CUI*s do not scale well.

*CUI*s do not take advantage of software leverage.

Make every program a filter.

Every program written
since the dawn of computing is a filter.

Programs do not create data – people do.

Computers convert data from one form to another.

Use stdin for data input.

Use stdout for data output.

Use stderr for out-of-band information.

Ten Lesser Tenets

Allow the User to tailor the environment.
Make operating system kernels small and lightweight.
Use lower case and keep it short.
Save Trees.
Silence is golden.
Think parallel.
The sum of the parts is greater than the whole.
Look for the 90 percent solution.
Worse is better.
Think hierarchically.

Small programs are easier to combine with other tools.

ASCII text is a common interchange format.

ASCII text is easily read and edited.

ASCII data files simplify the use of Unix text tools.

Shell scripts give you awesome leverage.

Automate everything.

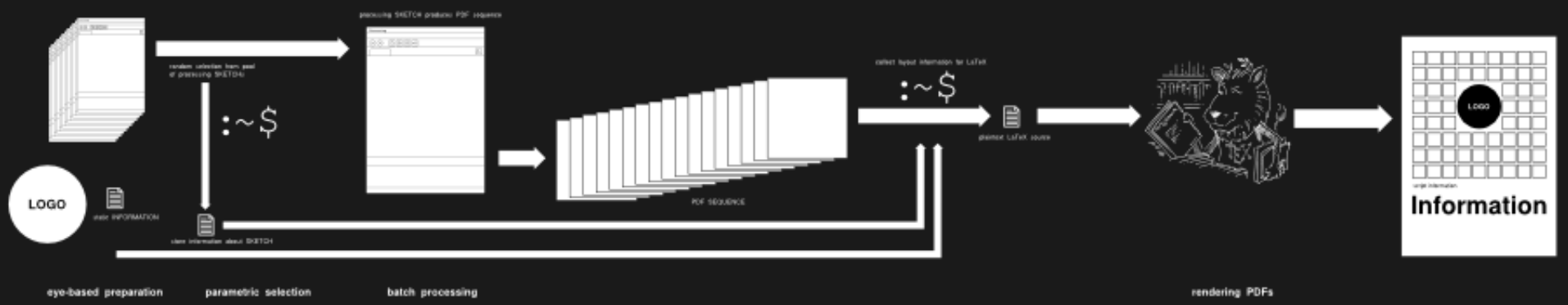
CUIs do not take advantage of software leverage.

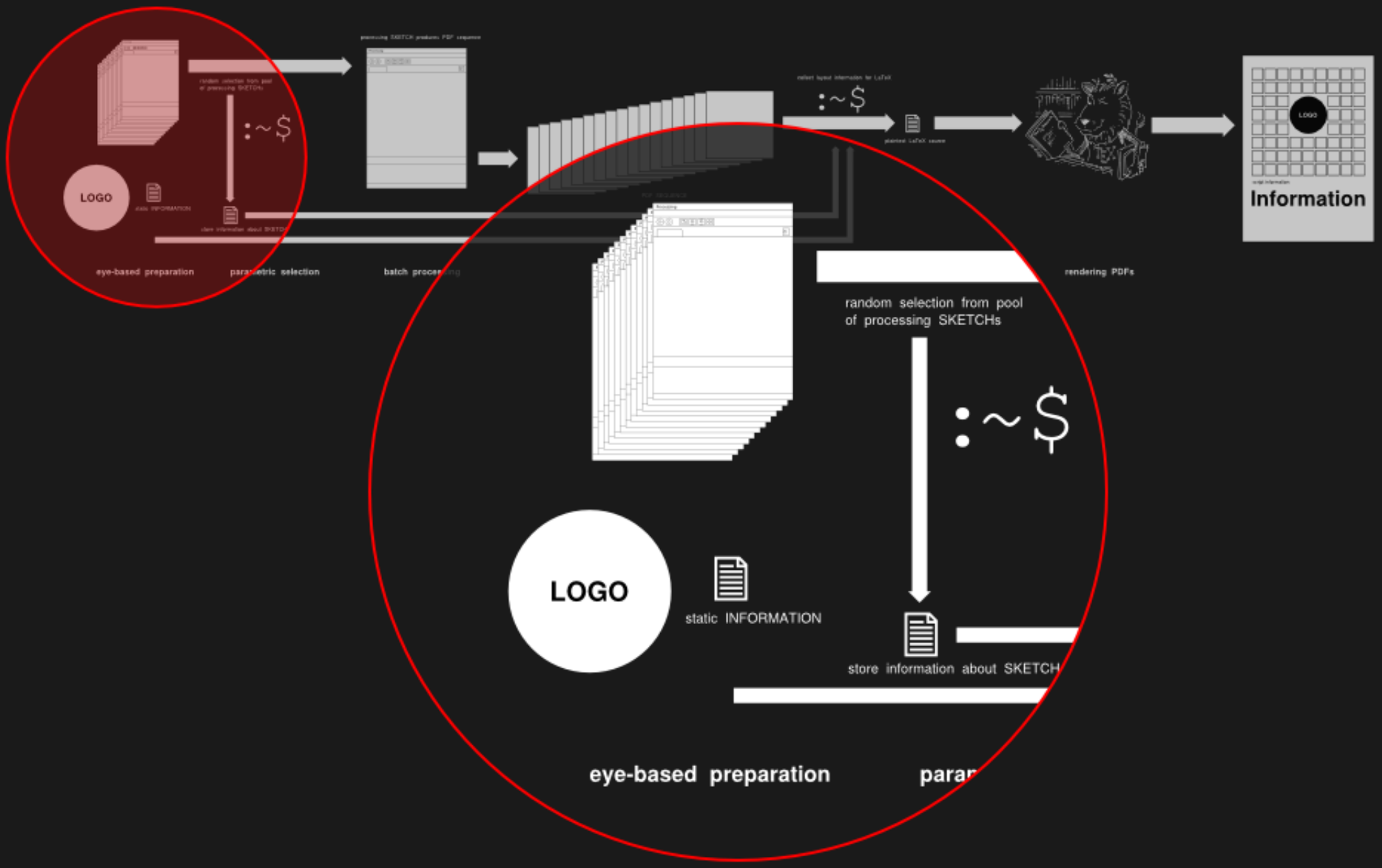
Don’t spend too much time making a program run faster.

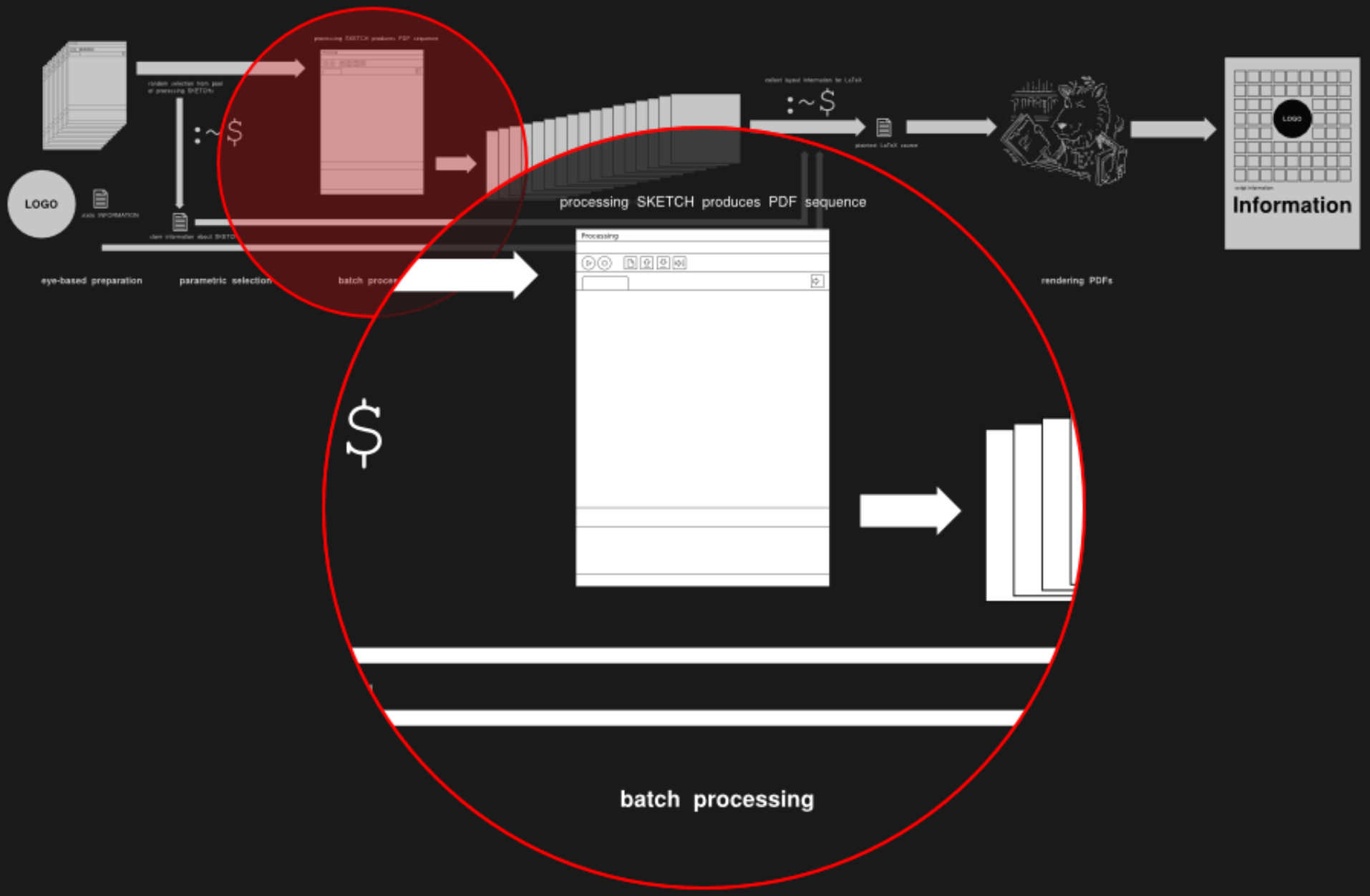
Avoid the not-invented-here syndrome.

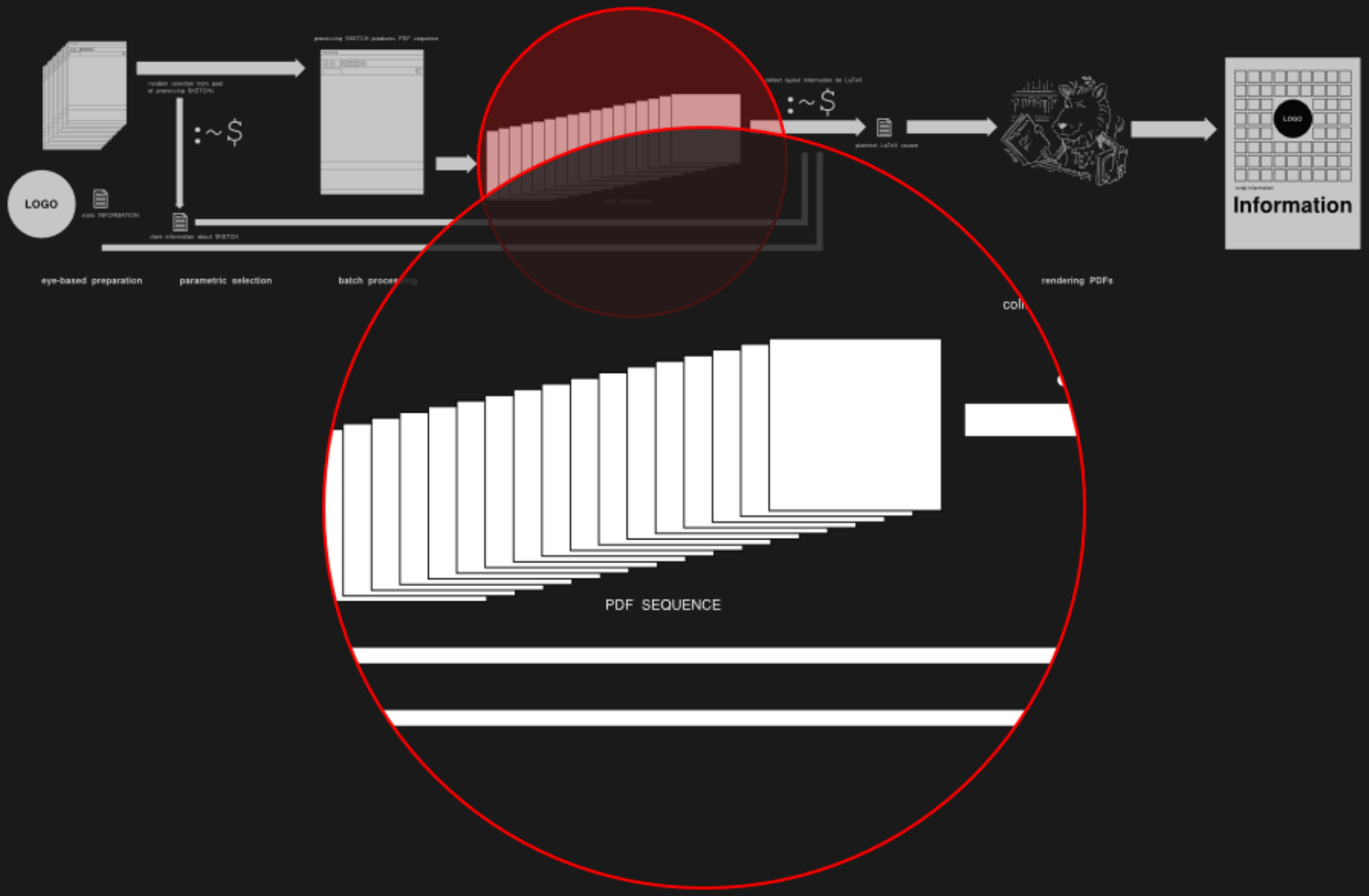
make art 2009

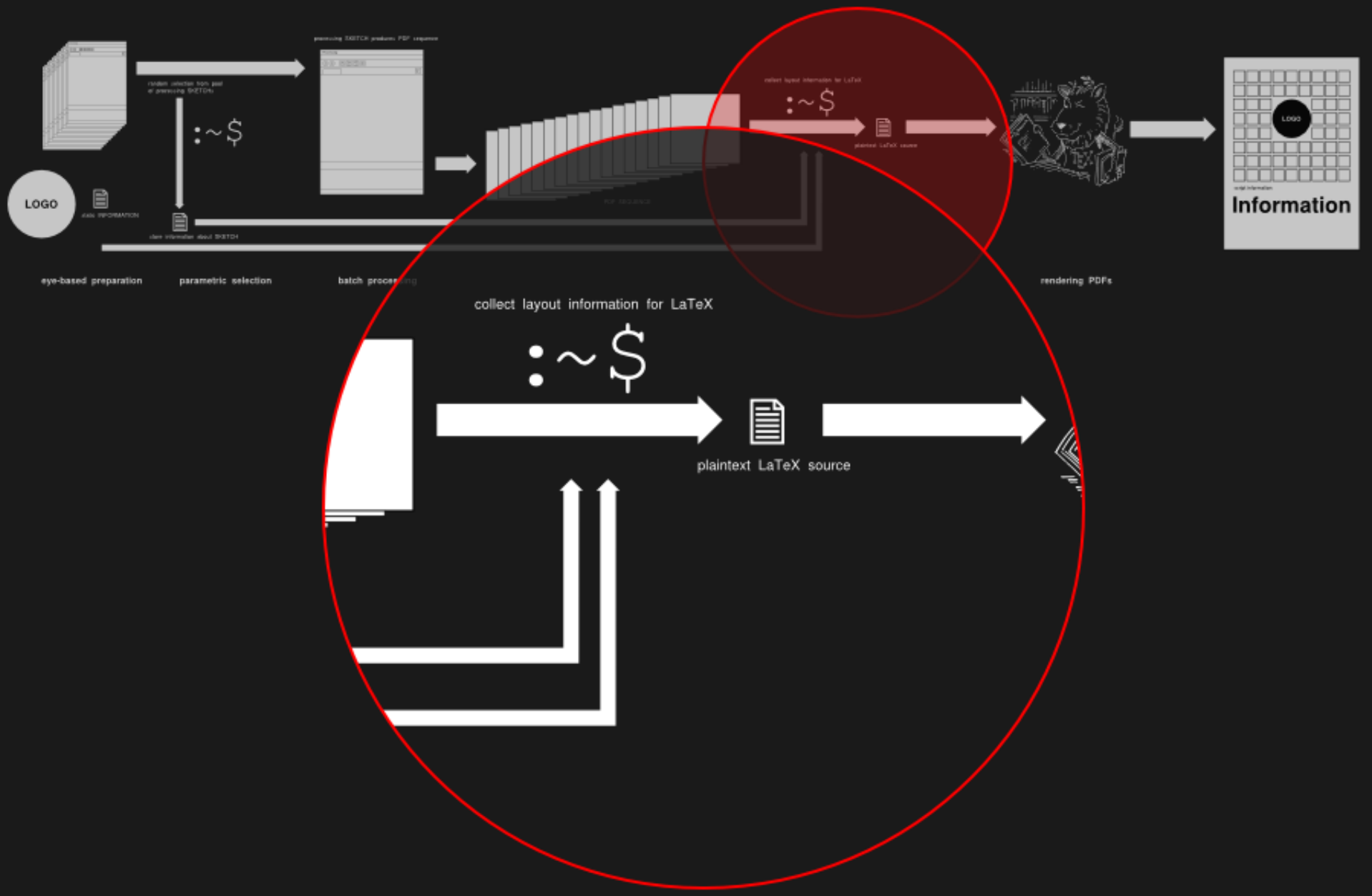
What the Fork?!

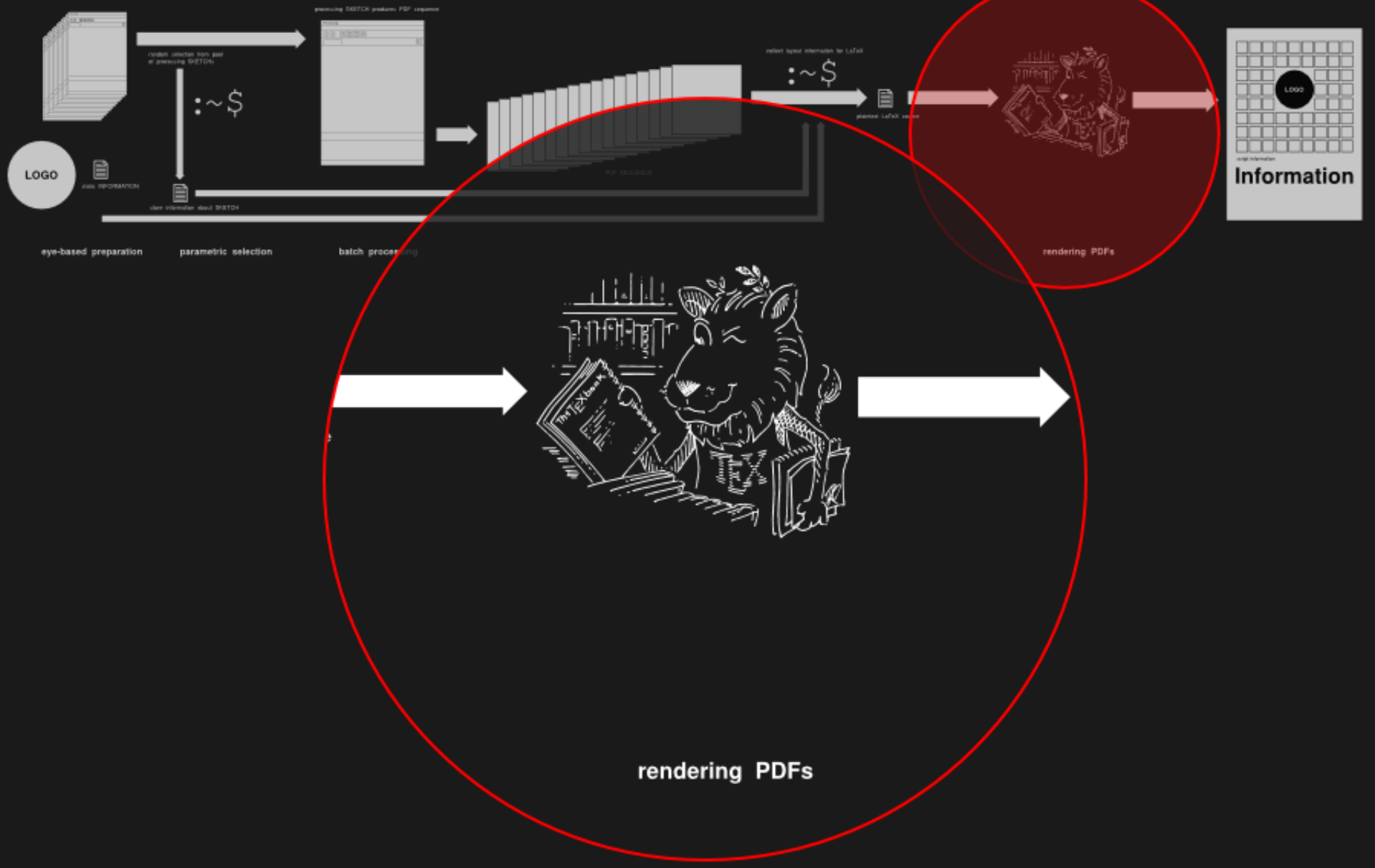

processing SKETCH produces PDF sequence
random selection from pool
of processing SKETCHes
:~$
LOGO
static INFORMATION
client information about SKETCH
PDF SEQUENCE
collect layout information for LaTeX
:~$
generate LaTeX source
LOGO
Information
eye-based preparation
parametric selection
batch processing
rendering PDFs

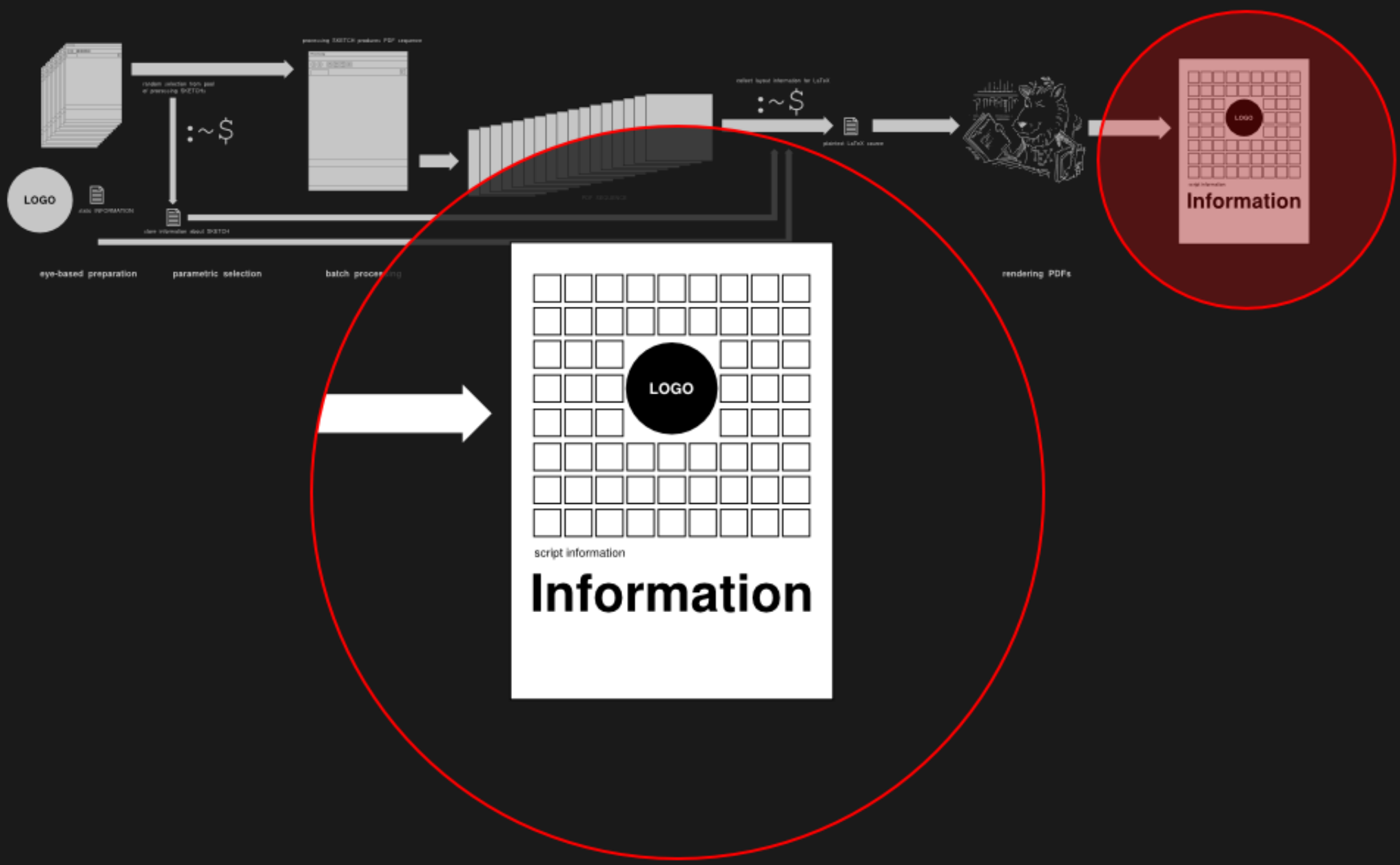

LOGO
eye-based preparation
rendering PDFs
Information
random selection from pool
of processing SKETCHs
LOGO
static INFORMATION
store information about SKETCH
eye-based preparation


LOGO
:~$
processing SKETCH produces PDF sequence
Processing
:~$
Information
LOGO
eye-based preparation
parametric selection
rendering PDFs
batch processing


LOGO
Information
eye-based preparation
parametric selection
batch proces
rendering PDFs
PDF SEQUENCE


LOGO
Information
eye-based preparation
parametric selection
rendering PDFs
collect layout information for LaTeX
:~$
plaintext LaTeX source


LOGO
:~$
:~$
Information
eye-based preparation
parametric selection
rendering PDFs
rendering PDFs

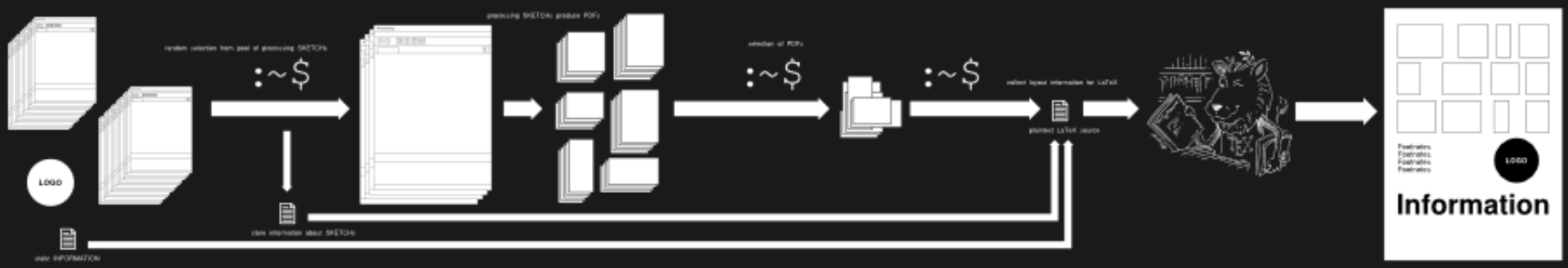

LOGO
:~$
:~$
LOGO
Information
eye-based preparation
parametric selection
rendering PDFs
LOGO
script information
Information

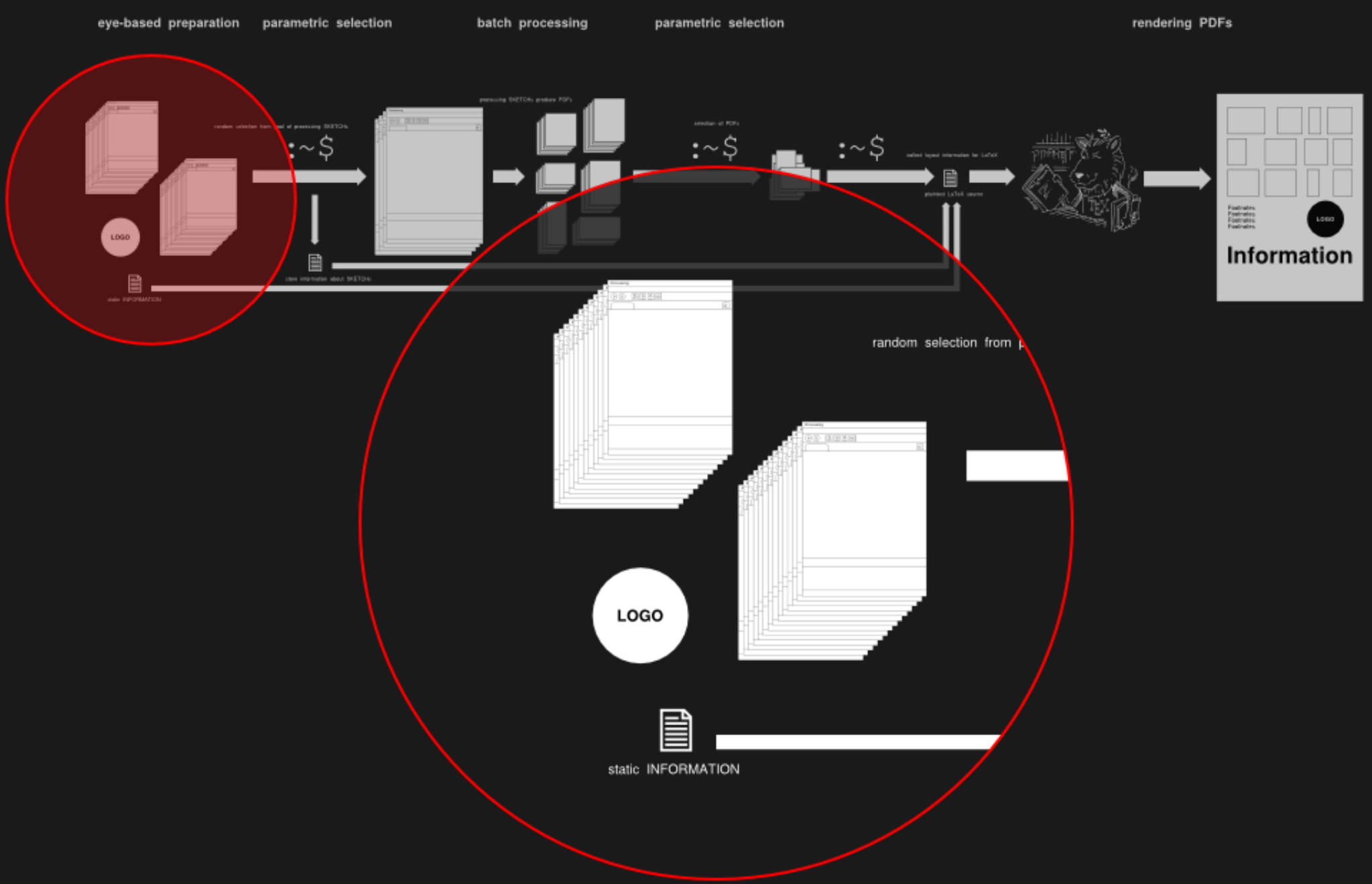
http://makeart.goto10.org/2009/

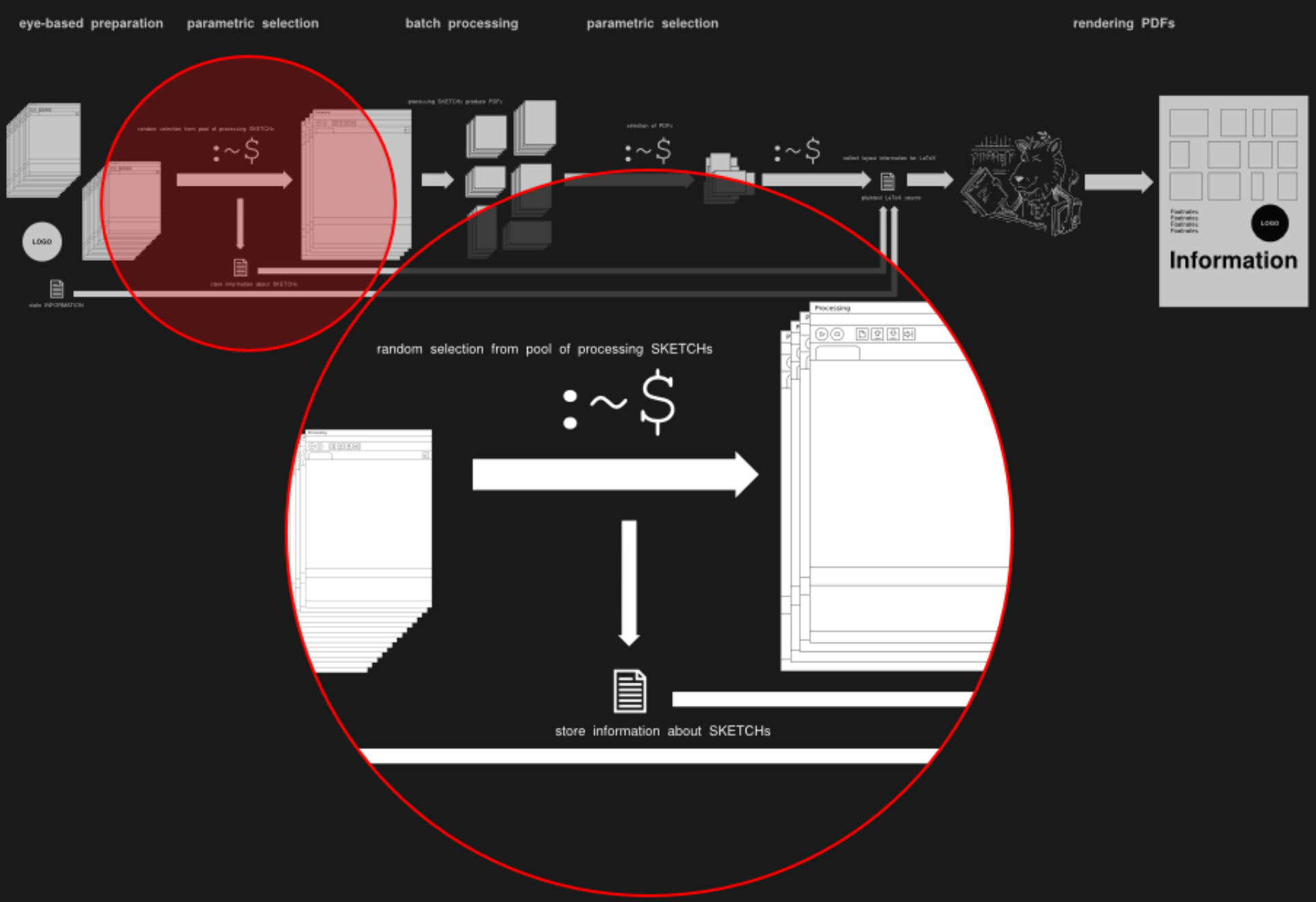
http://www.forkable.eu/generators/wtf

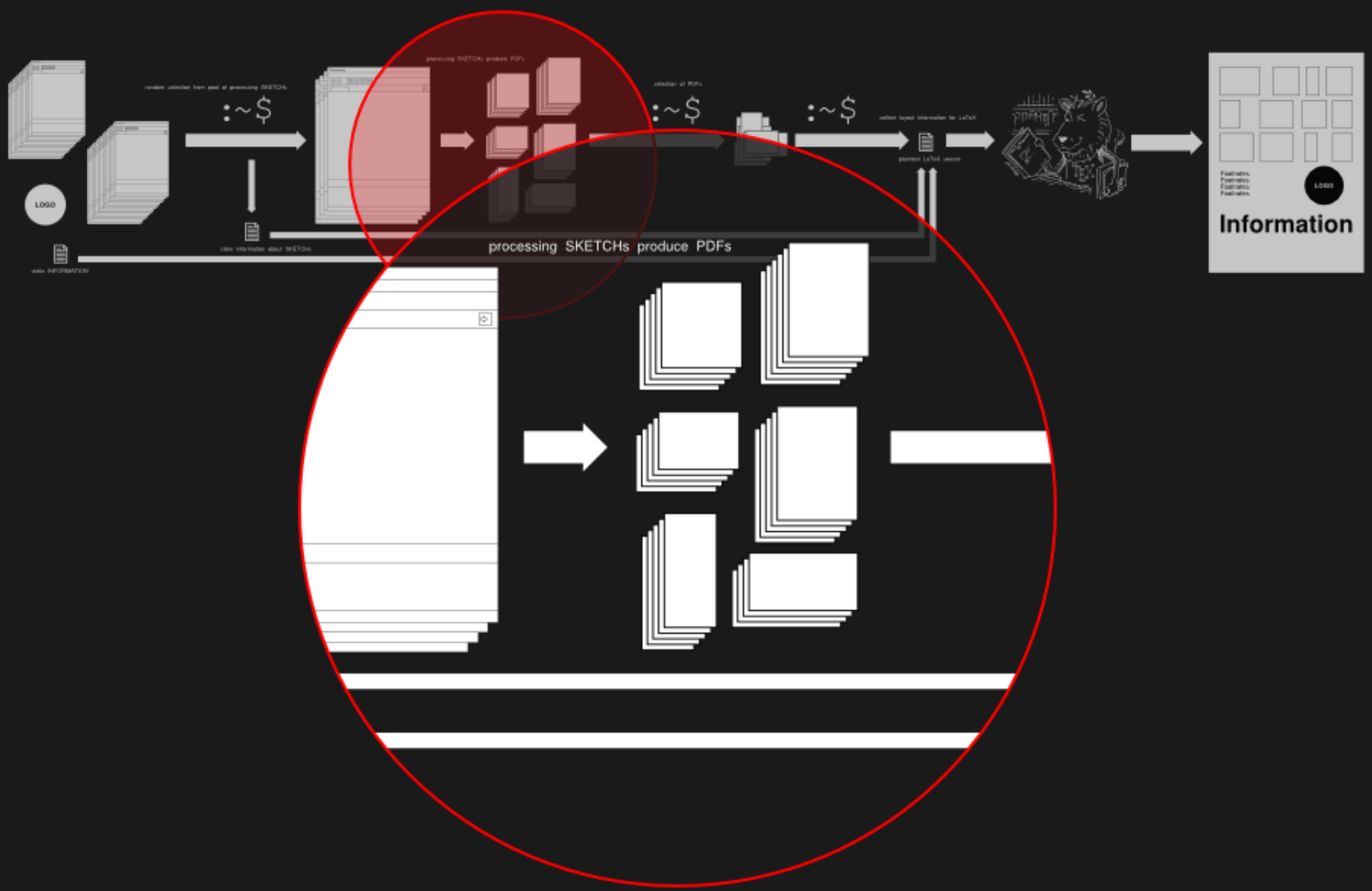
~~make art~~ 2010

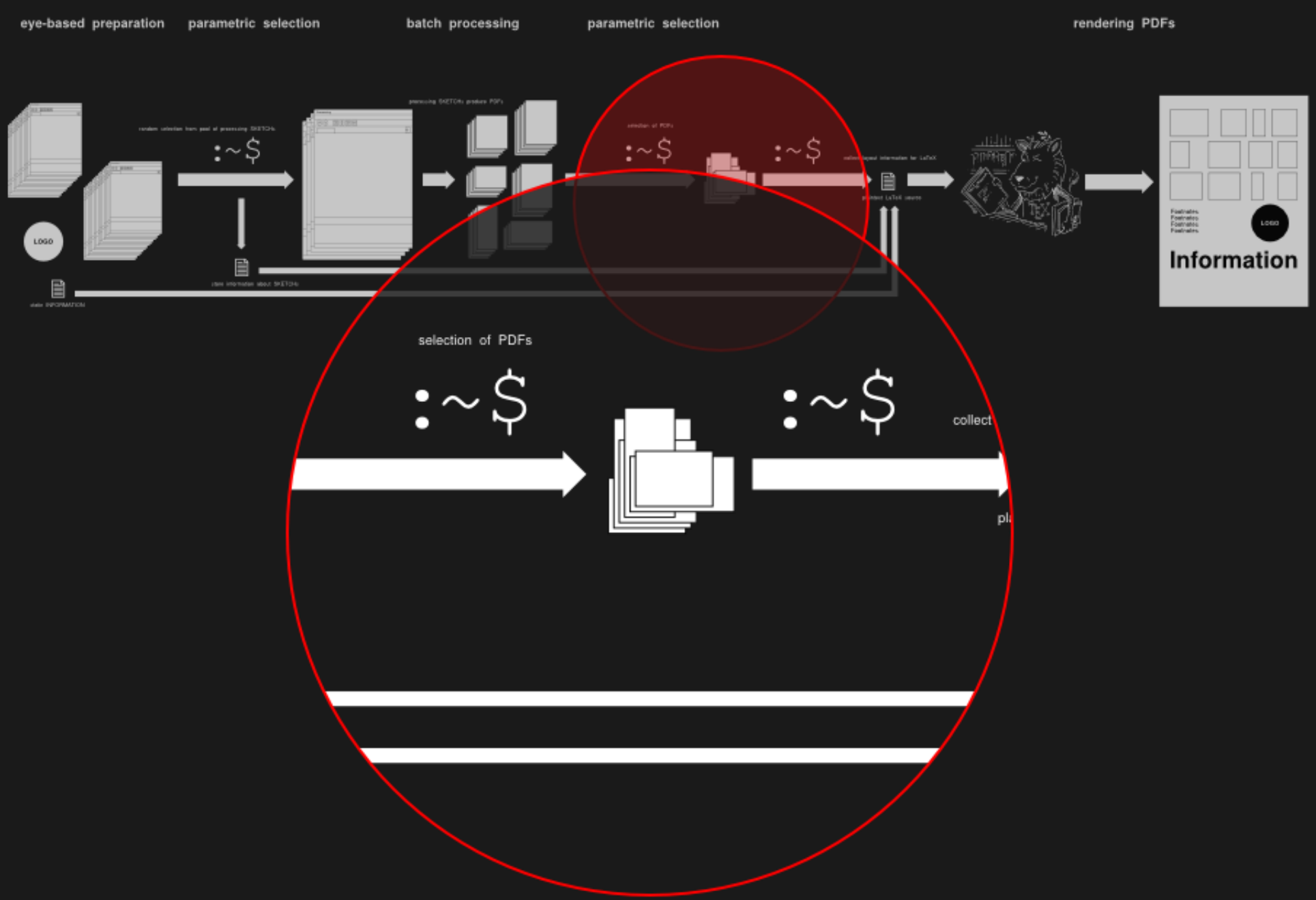
chmod +x

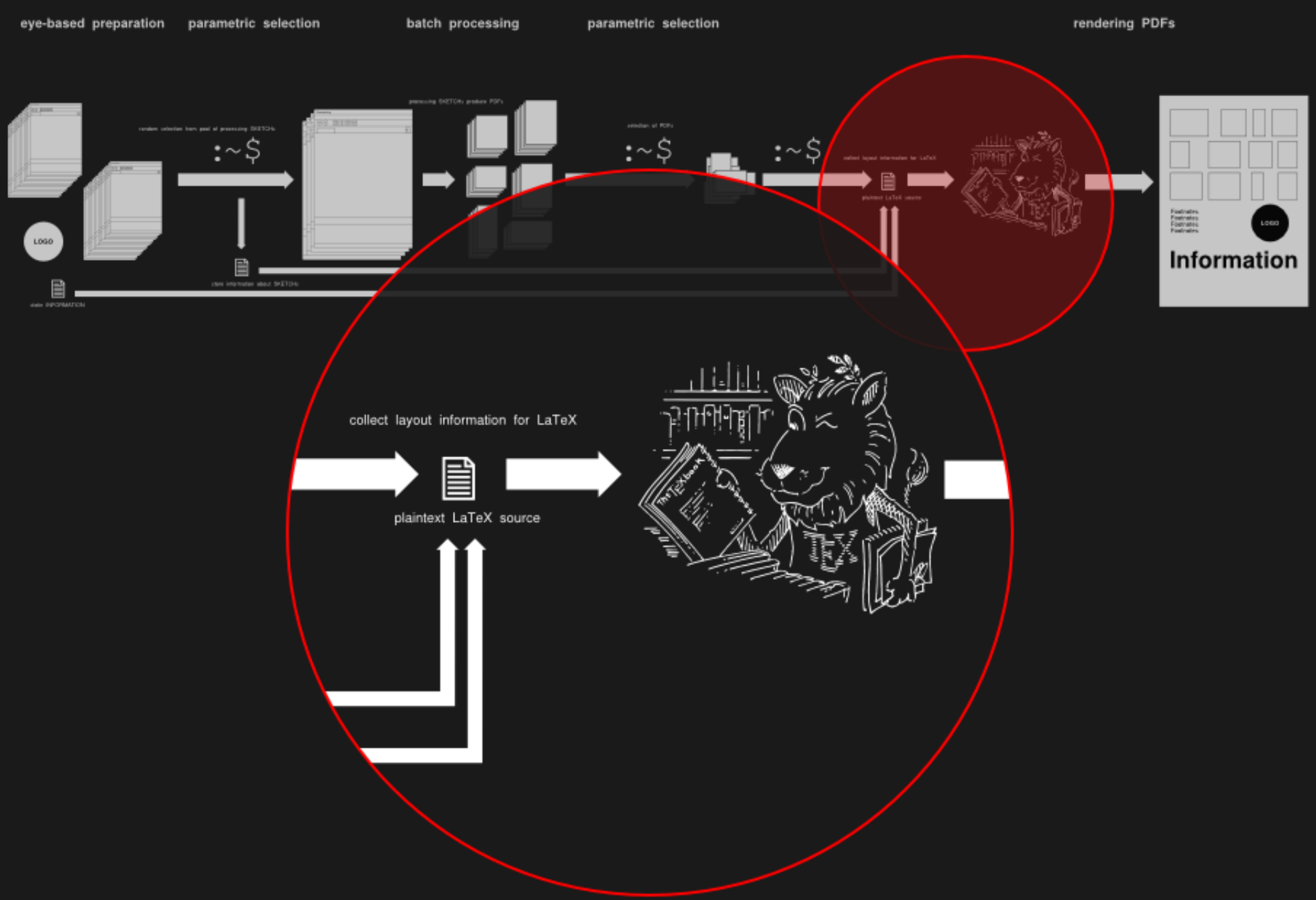

eye-based preparation
parametric selection
batch processing
parametric selection
rendering PDFs
LOGO
:~$
:~$
:~$
Information
LOGO

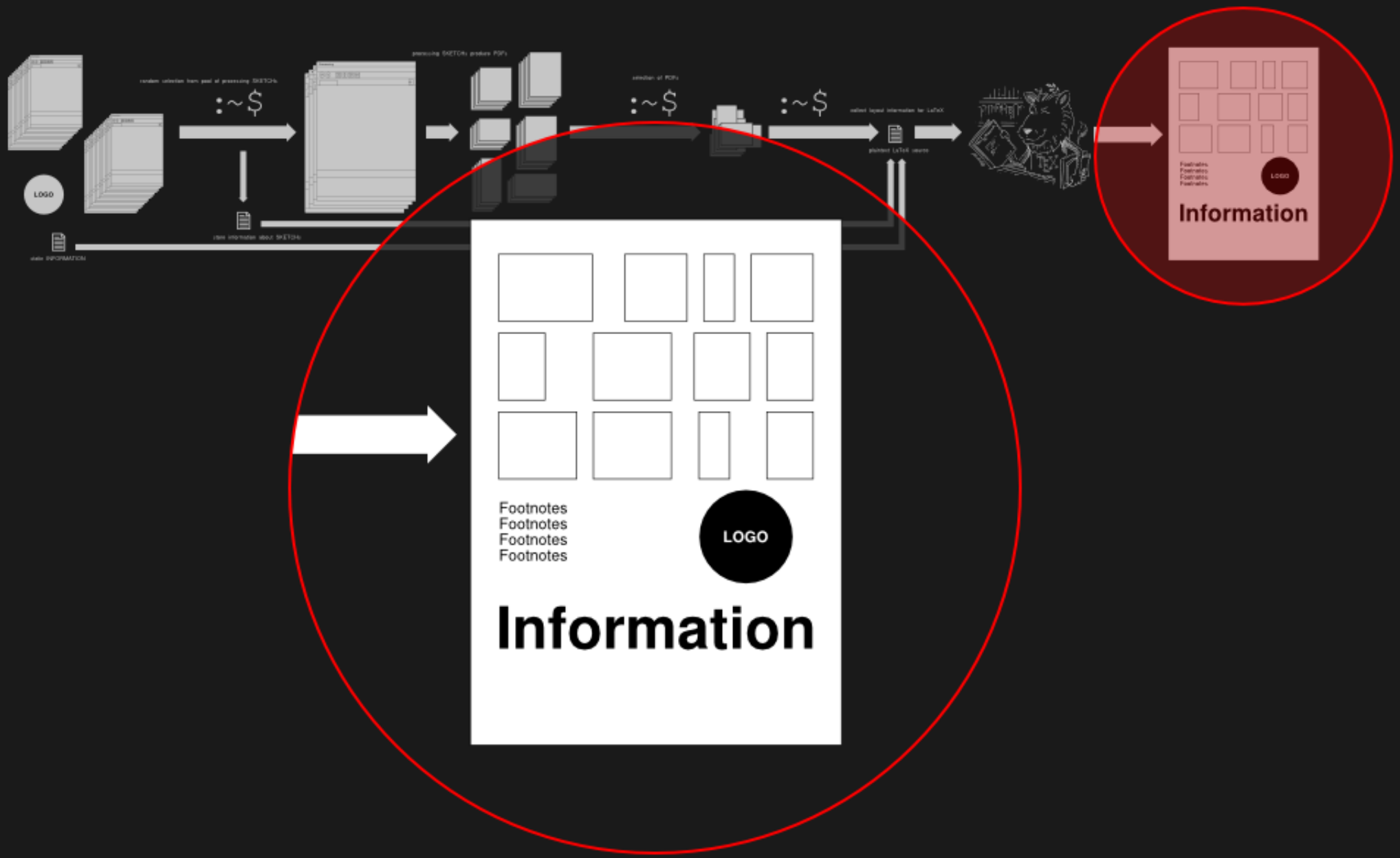

eye-based preparation
parametric selection
batch processing
parametric selection
rendering PDFs
LOGO
:~$
:~$
:~$
Information
random selection from
LOGO
static INFORMATION


eye-based preparation
parametric selection
batch processing
parametric selection
rendering PDFs
LOGO
:~$
Information
random selection from pool of processing SKETCHs
:~$
store information about SKETCHs


eye-based preparation
parametric selection
batch processing
parametric selection
rendering PDFs
LOGO
Information
processing SKETCHs produce PDFs


eye-based preparation
parametric selection
batch processing
parametric selection
rendering PDFs
LOGO
:~$
Information
selection of PDFs
:~$
:~$
collect


eye-based preparation
parametric selection
batch processing
parametric selection
rendering PDFs
LOGO
:~$
:~$
:~$
Information
collect layout information for LaTeX
plaintext LaTeX source

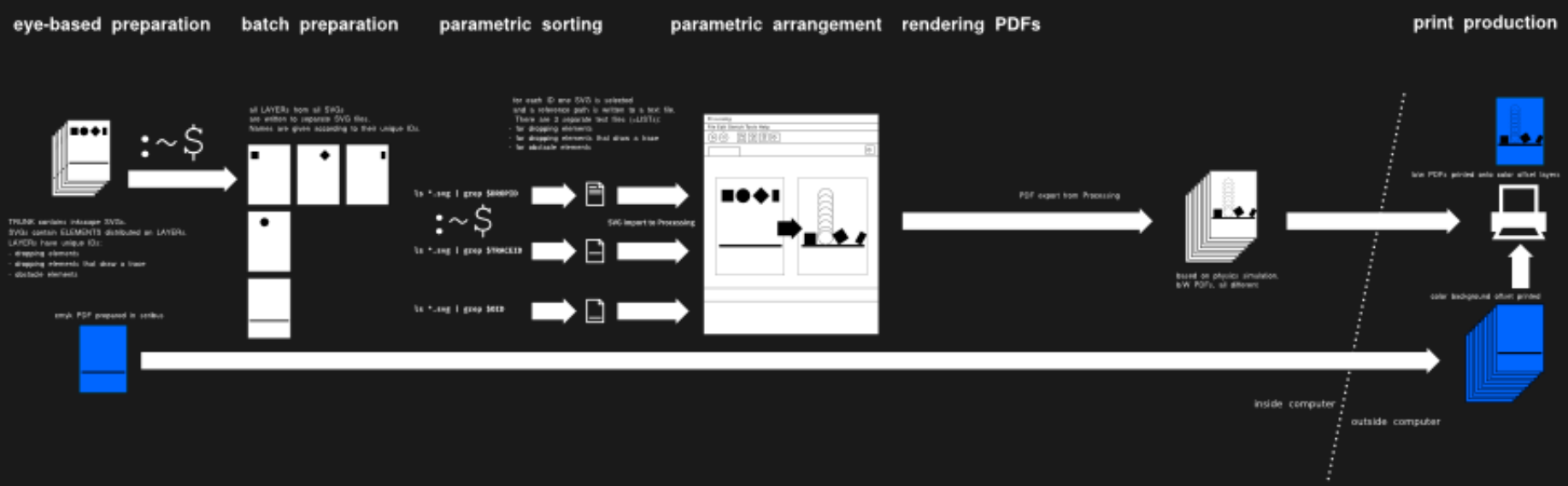

eye-based preparation
parametric selection
batch processing
parametric selection
rendering PDFs
LOGO
:~$
:~$
:~$
Footnotes
Footnotes
Footnotes
Footnotes
LOGO
Information

http://makeart.goto10.org/chmod+x/

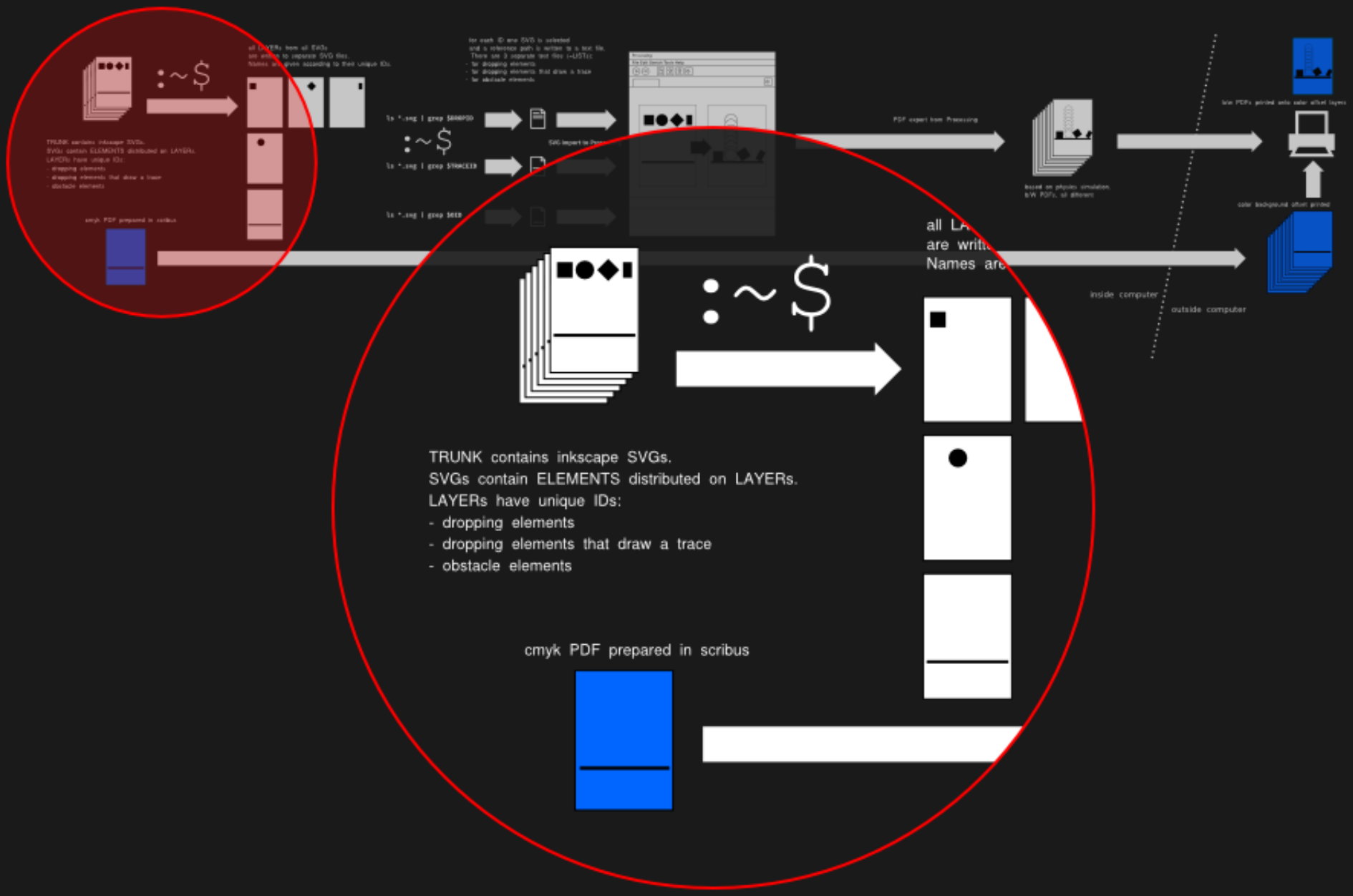
http://www.forkable.eu/generators/chmod+x

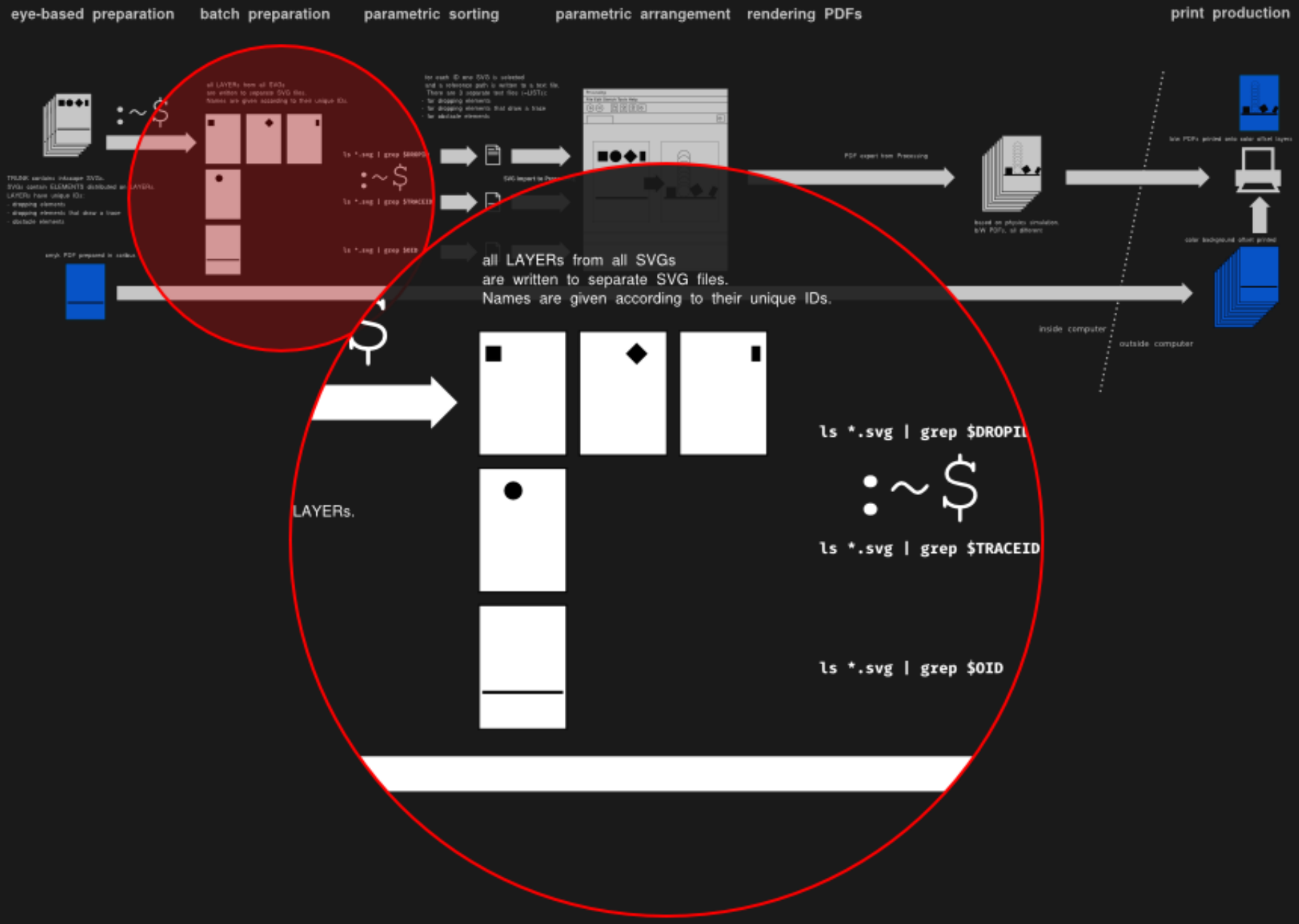
make art 2010

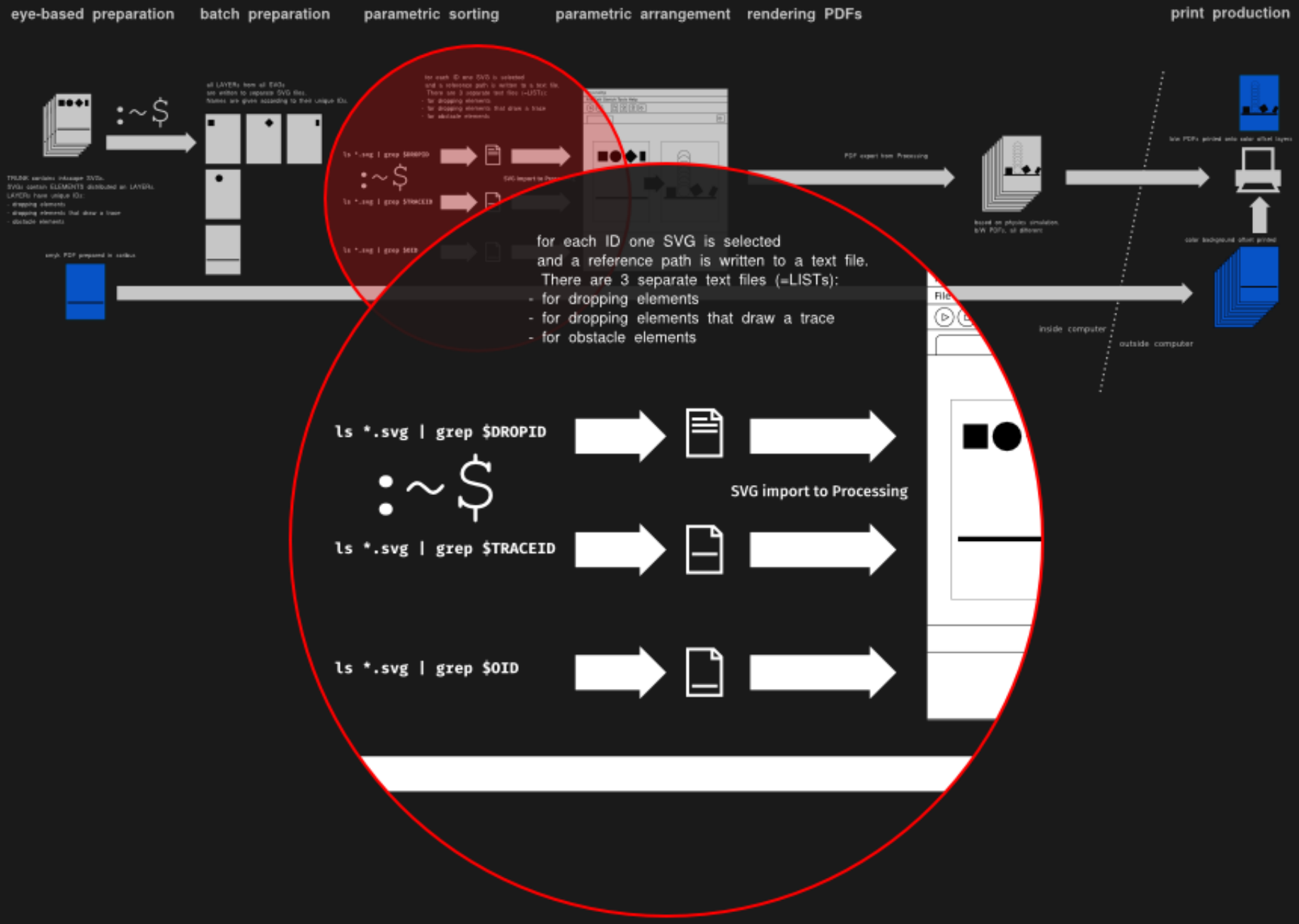
in-Between Design


eye-based preparation
batch preparation
parametric sorting
parametric arrangement
rendering PDFs
print production
:~$
:~$
SVG import to Processing
PDF export from Processing
inside computer
outside computer

eye-based preparation batch preparation parametric sorting parametric arrangement rendering PDFs print production

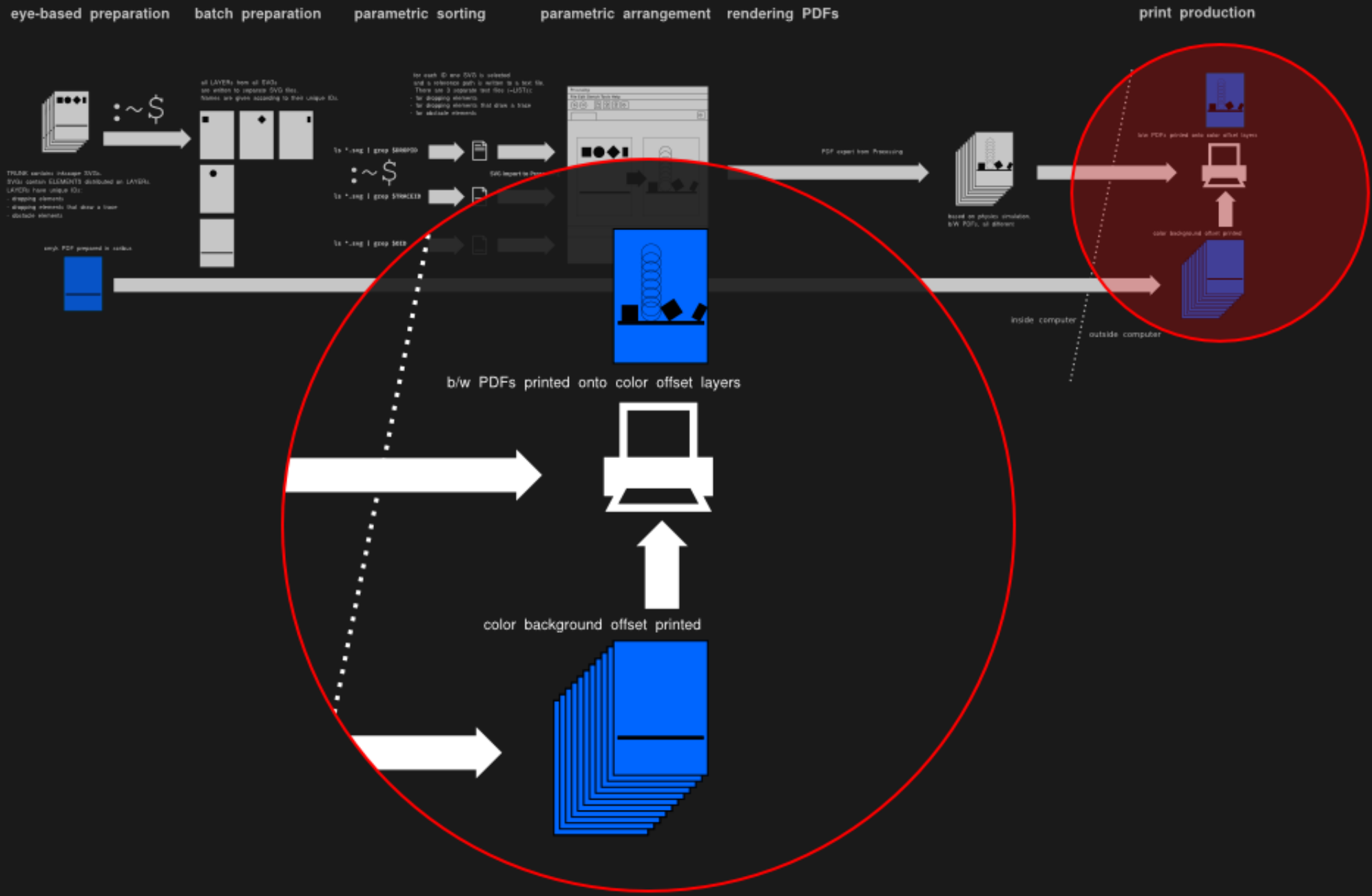



eye-based preparation
batch preparation
parametric sorting
parametric arrangement
rendering PDFs
print production
all LAYERs from all SVGs
are written to separate SVG files.
Names are given according to their unique IDs.
LAYERs.
ls *.svg | grep $DROPID
ls *.svg | grep $TRACEID
ls *.svg | grep $OID
inside computer
outside computer


eye-based preparation
batch preparation
parametric sorting
parametric arrangement
rendering PDFs
print production
for each ID one SVG is selected
and a reference path is written to a text file.
There are 3 separate text files (=LISTs):
- for dropping elements
- for dropping elements that draw a trace
- for obstacle elements
ls *.svg | grep $DROPID
ls *.svg | grep $TRACEID
ls *.svg | grep $OID
SVG import to Processing
inside computer
outside computer


eye-based preparation
batch preparation
parametric sorting
parametric arrangement
rendering PDFs
print production
inside computer
outside computer
SVG import to Processing
Processing
File Edit Sketch Tools Help

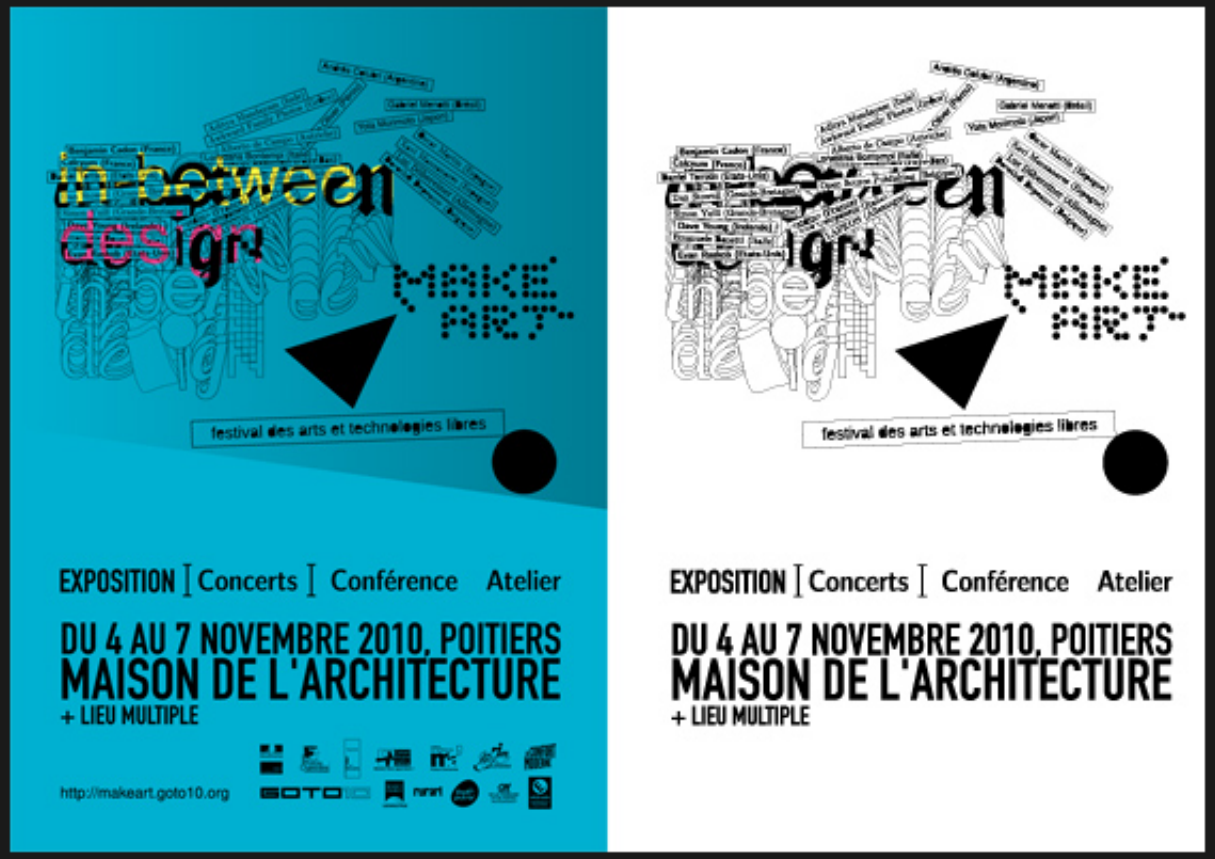
eye-based preparation batch preparation parametric sorting parametric arrangement rendering PDFs print production


eye-based preparation
batch preparation
parametric sorting
parametric arrangement
rendering PDFs
print production
b/w PDFs printed onto color offset layers
color background offset printed
inside computer
outside computer

in-between design

http://makeart.goto10.org


in-between design
MAKE ART
festival des arts et technologies libres
Exposition | CONCERTS | CONFÉRENCES Atelier
du 4 au 7 novembre 2010, Poitiers
MAISON DE L'ARCHITECTURE
+ LIEU MULTIPLE
http://makeart.goto10.org
GOTO10
in-between design
MAKE ART
festival des arts et technologies libres
Exposition | CONCERTS | CONFÉRENCES Atelier
du 4 au 7 novembre 2010, Poitiers
MAISON DE L'ARCHITECTURE
+ LIEU MULTIPLE


in-between
design
MAKE
ART
festival des arts et technologies libres
EXPOSITION | Concerts | Conférence Atelier
DU 4 AU 7 NOVEMBRE 2010, POITIERS
MAISON DE L'ARCHITECTURE
+ LIEU MULTIPLE
http://makeart.goto10.org
GOTO10
in-between
design
MAKE
ART
festival des arts et technologies libres
EXPOSITION | Concerts | Conférence Atelier
DU 4 AU 7 NOVEMBRE 2010, POITIERS
MAISON DE L'ARCHITECTURE
+ LIEU MULTIPLE

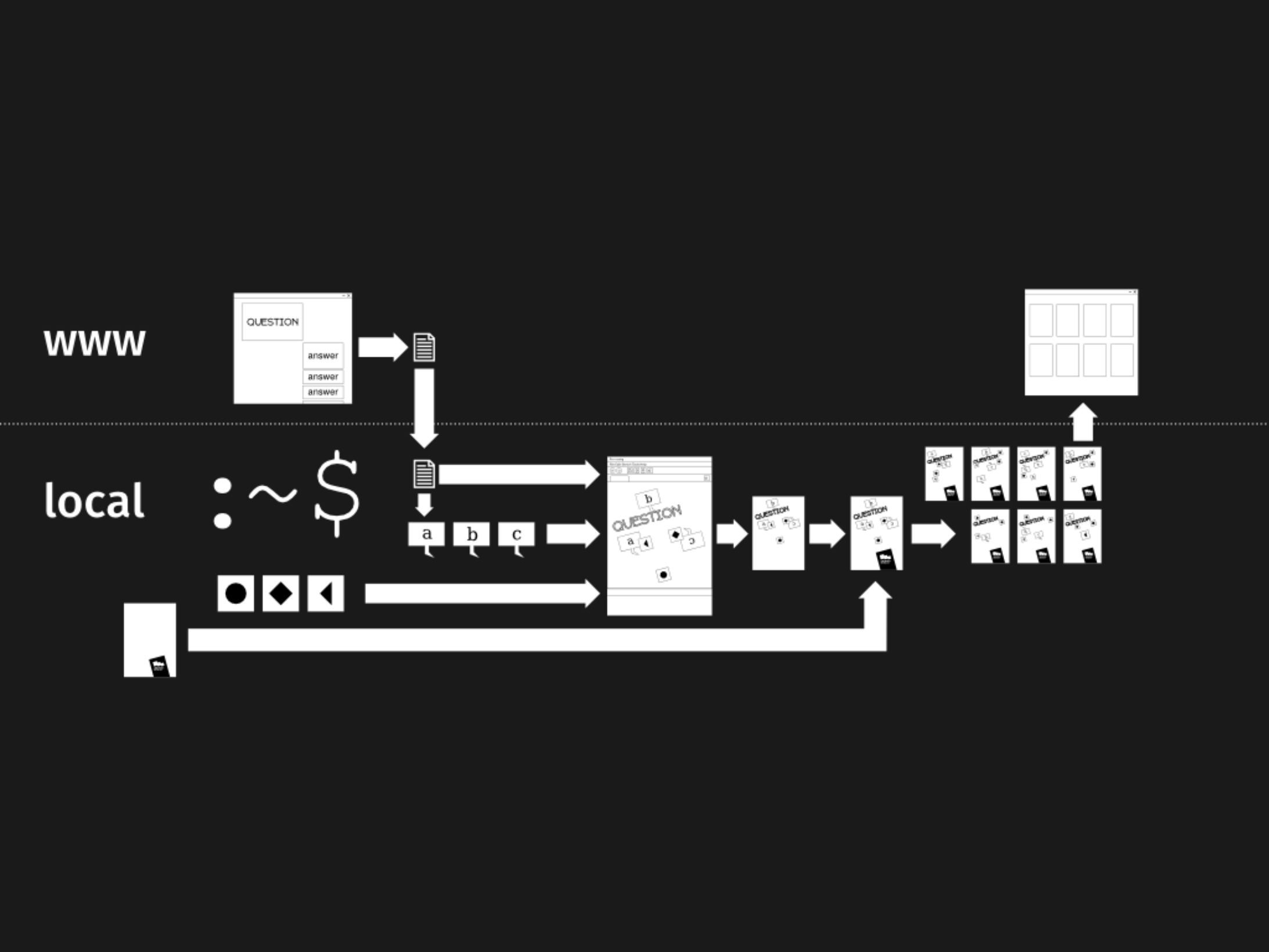

in-between
design
MAKE
ART
festival des arts et technologies libres
Exposition | Concerts | Conférence Atelier
DU 4 AU 7 NOVEMBRE 2010, POITIERS
MAISON DE L'ARCHITECTURE
+ LIEU MULTIPLE
http://makeart.goto10.org
GOTO

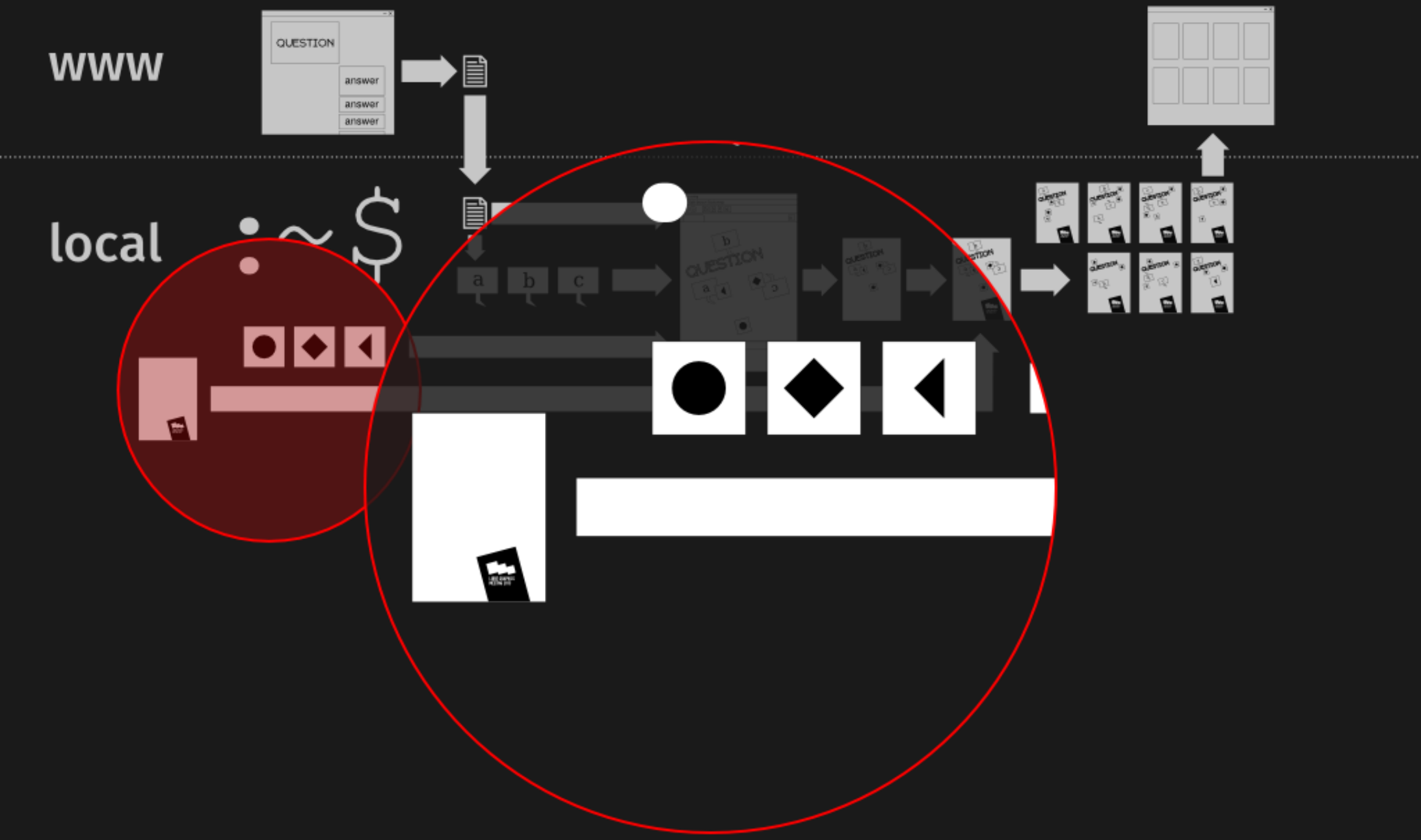
http://www.forkable.eu/generators/i-bd/o/non-free/fr/A3/recto/SEEME

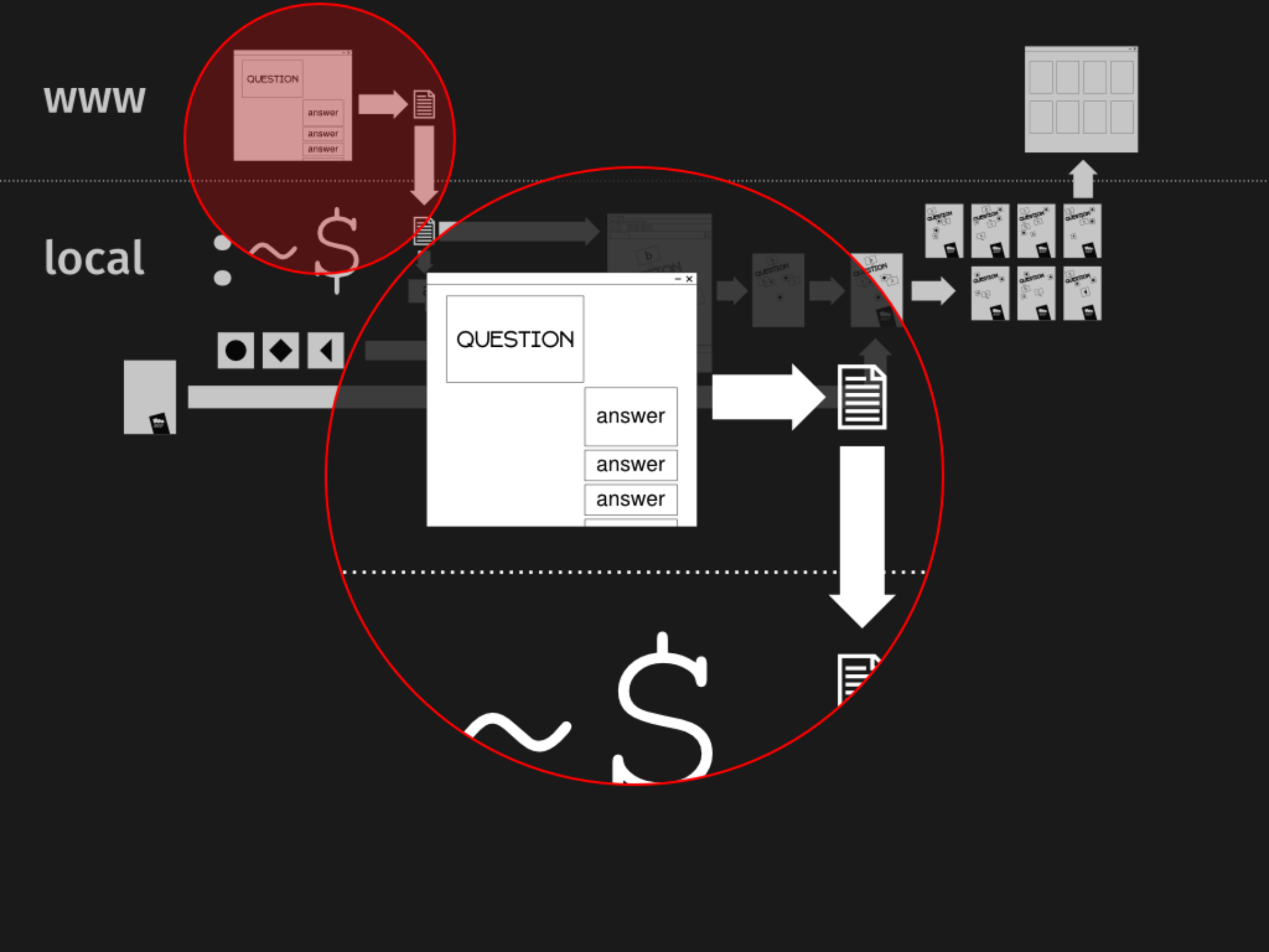
http://www.forkable.eu/generators/i-bd

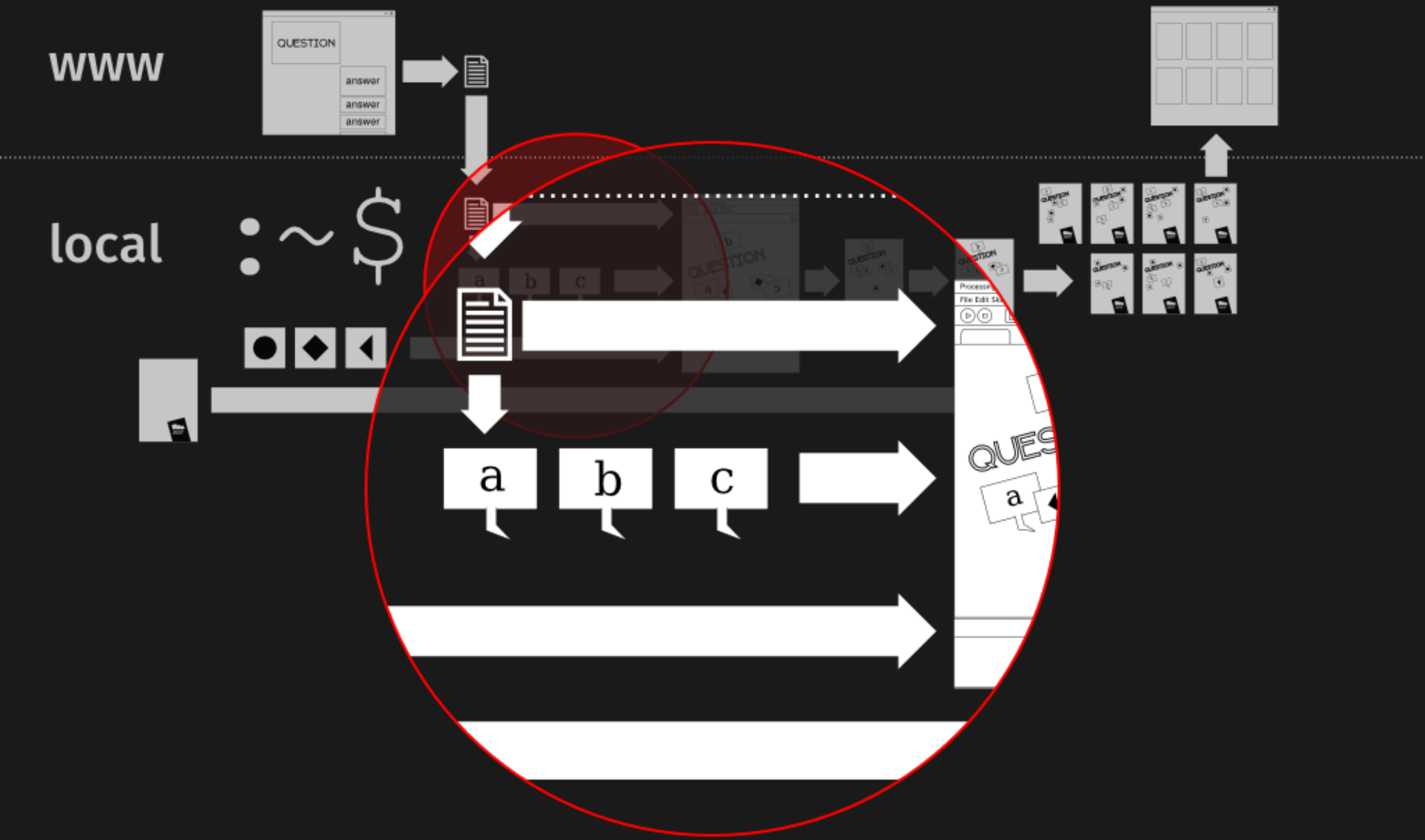
Libre Graphics Meeting 2013

Future Tools

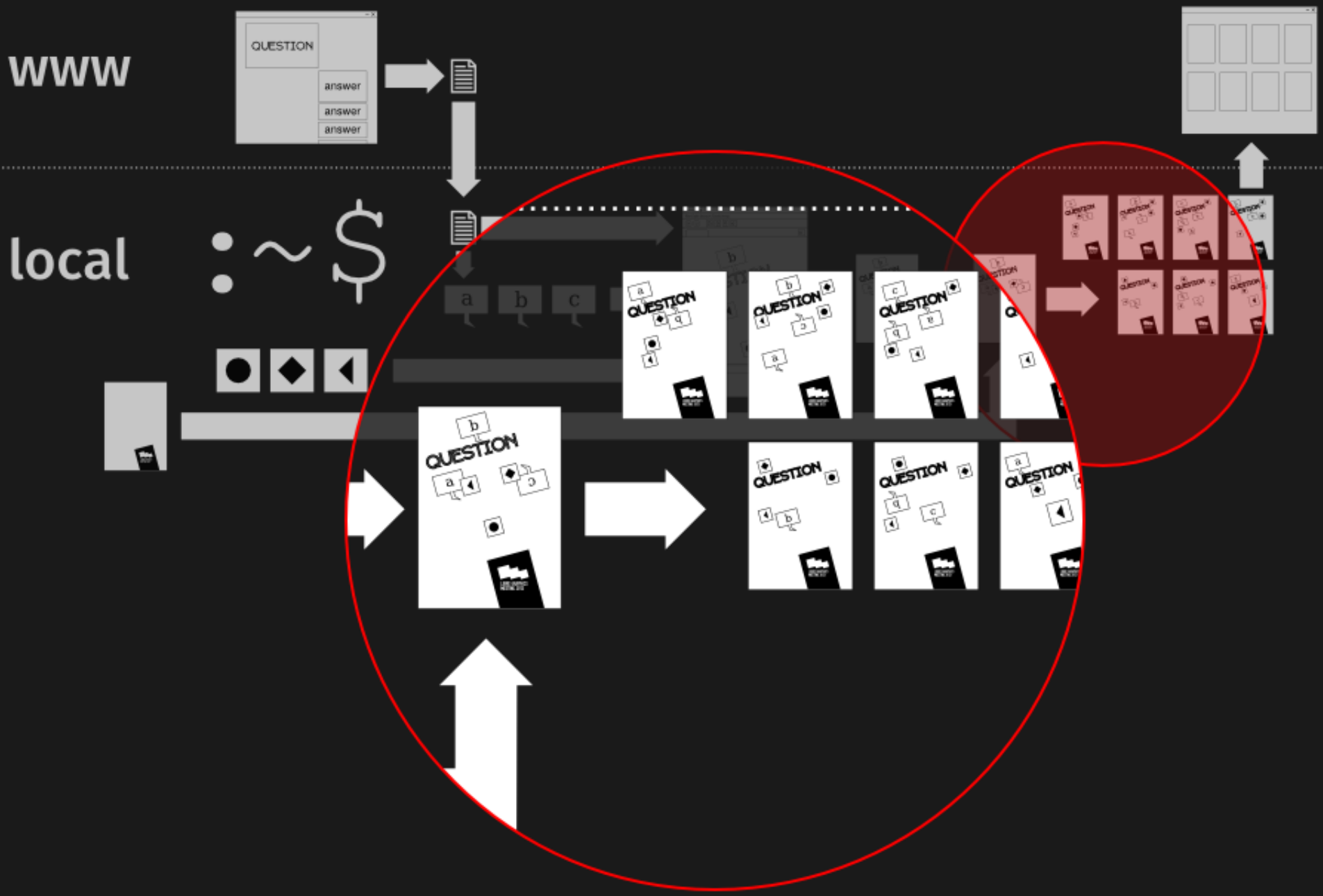

www
QUESTION
answer
answer
answer
local
:~$
a
b
c
b
QUESTION
a
c

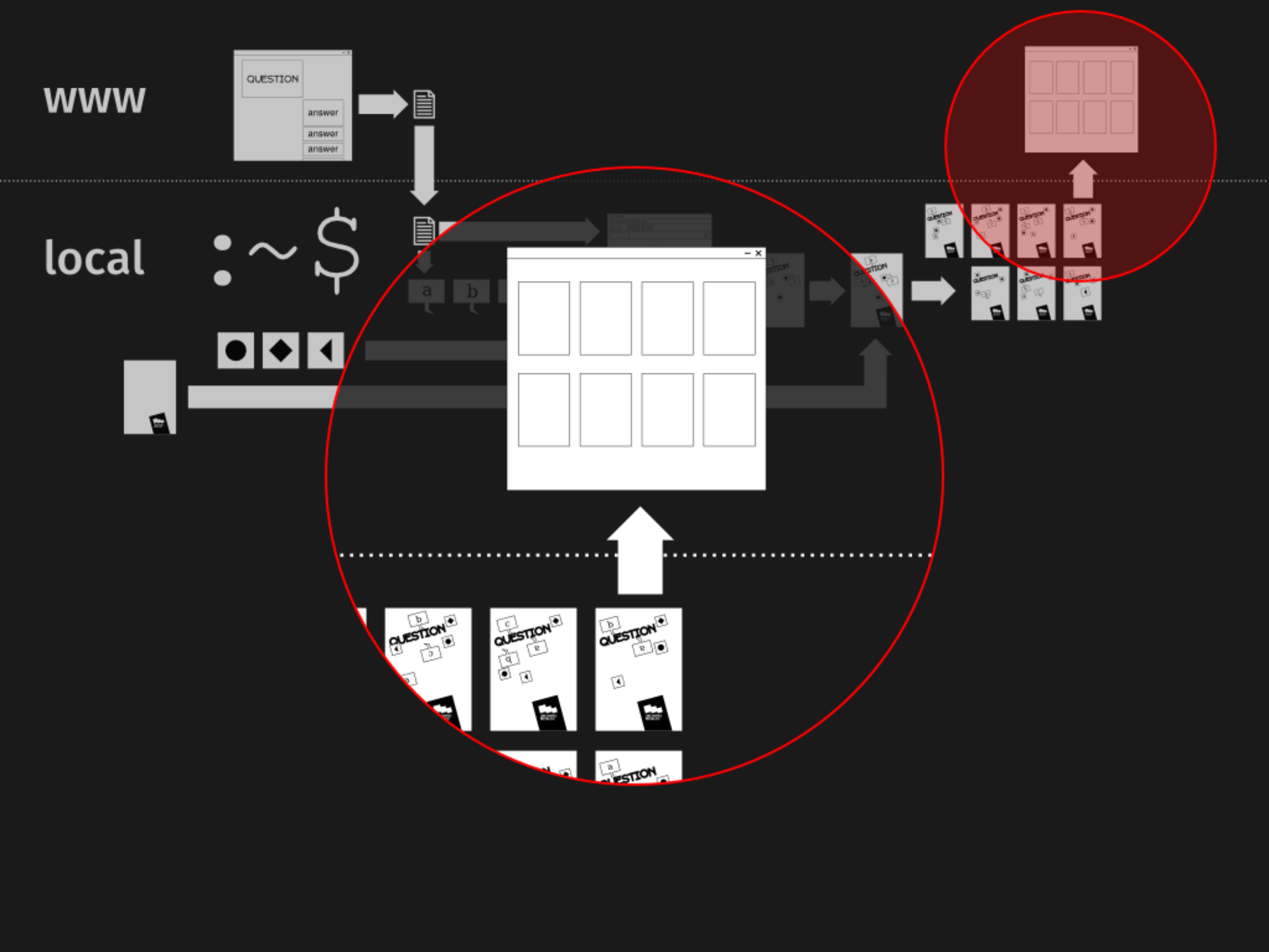

www
QUESTION
answer
answer
answer
local
:~$
a
b
c
QUESTION


www
local
QUESTION
answer
answer
answer
QUESTION
answer
answer
answer


www
QUESTION
answer
answer
answer
local
:~$
a
b
c
a
b
c


www
QUESTION
answer
answer
answer
local
:~$
Processing
file Edit Sketch Tools Help
b
QUESTION
a
c
b
QUESTION
a
c

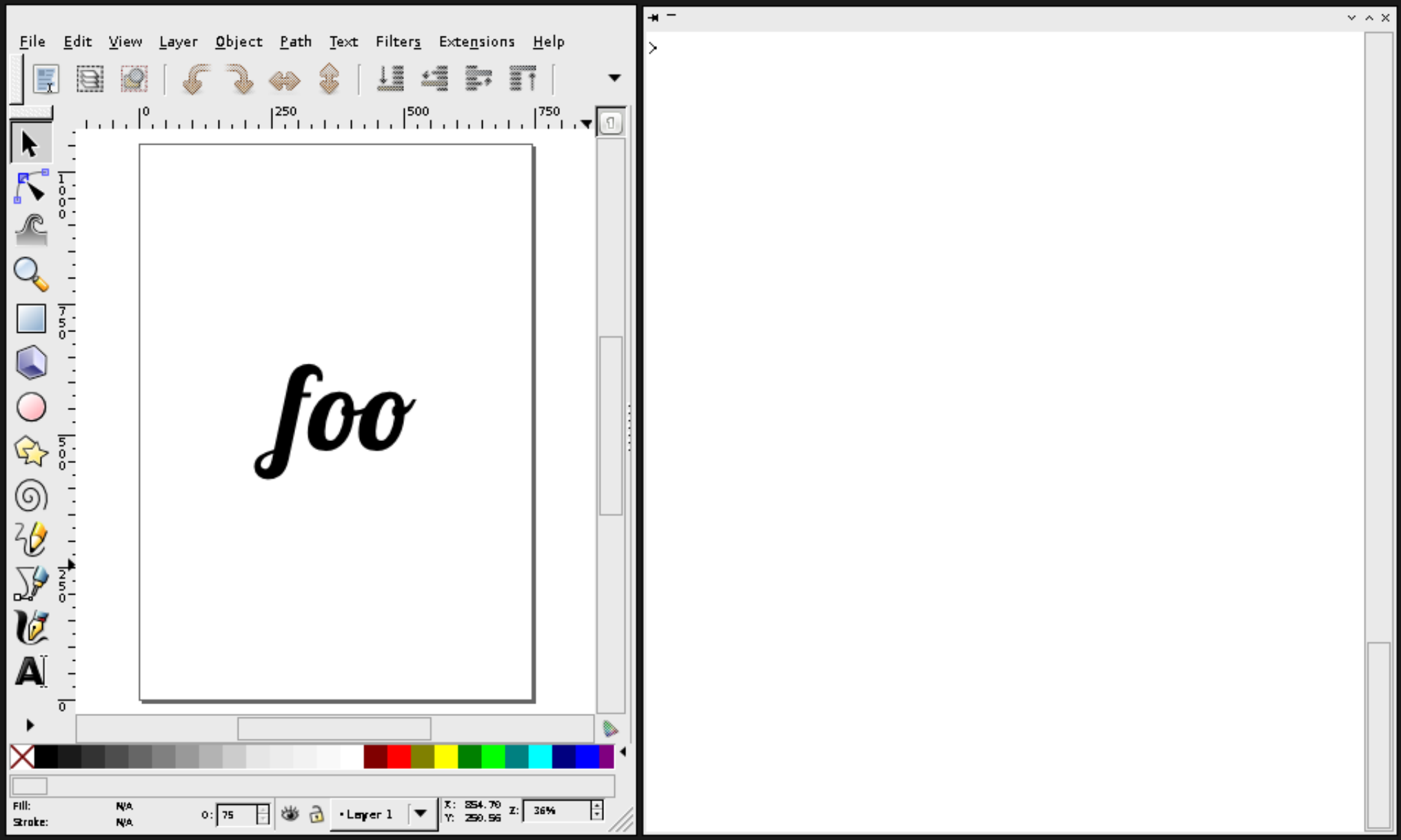

www
QUESTION
answer
answer
answer
local
:~$
a
b
c
QUESTION

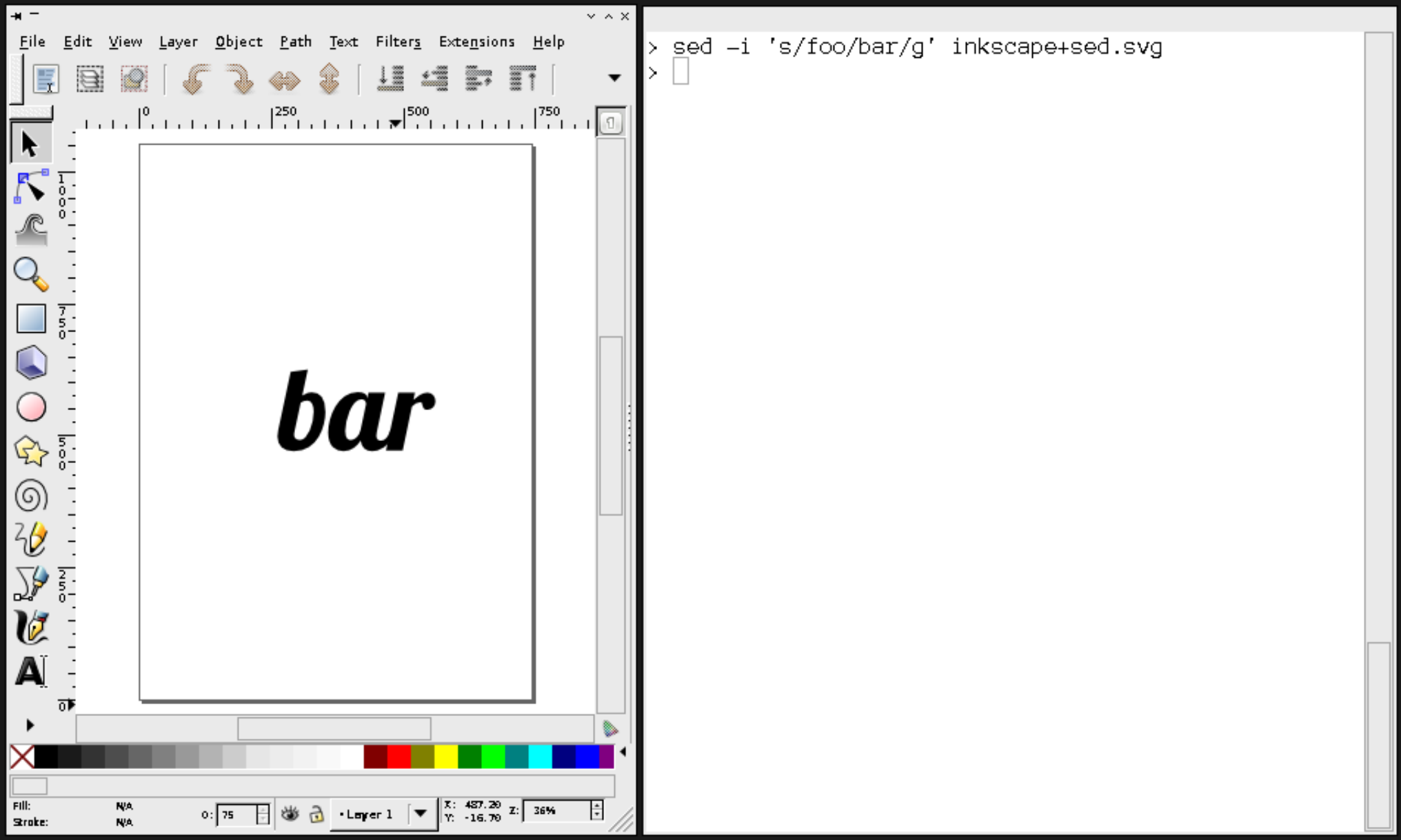

www
QUESTION
answer
answer
answer
local
:~$
a
b
QUESTION
QUESTION
QUESTION

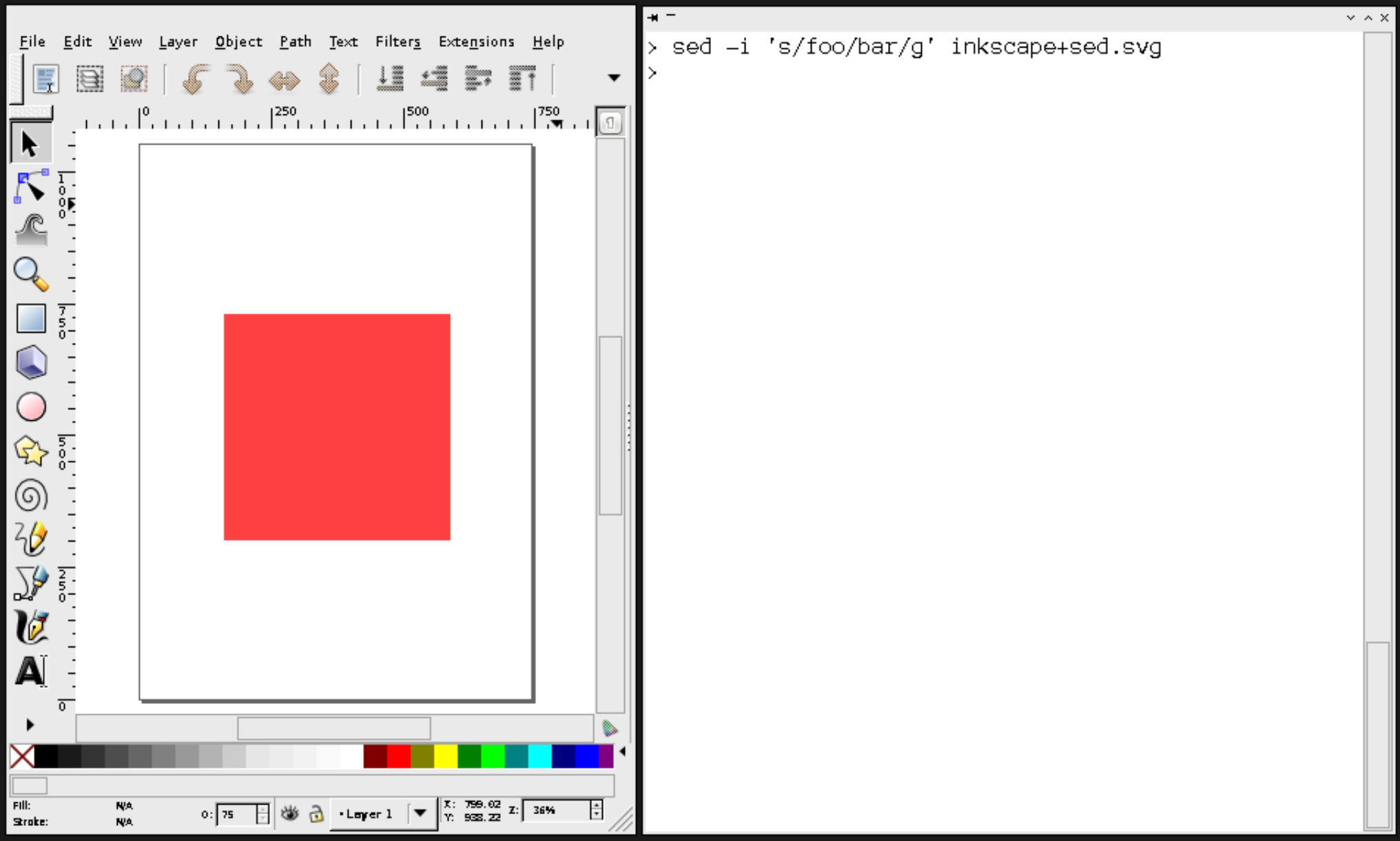
http://www.forkable.eu/generators/r+w/

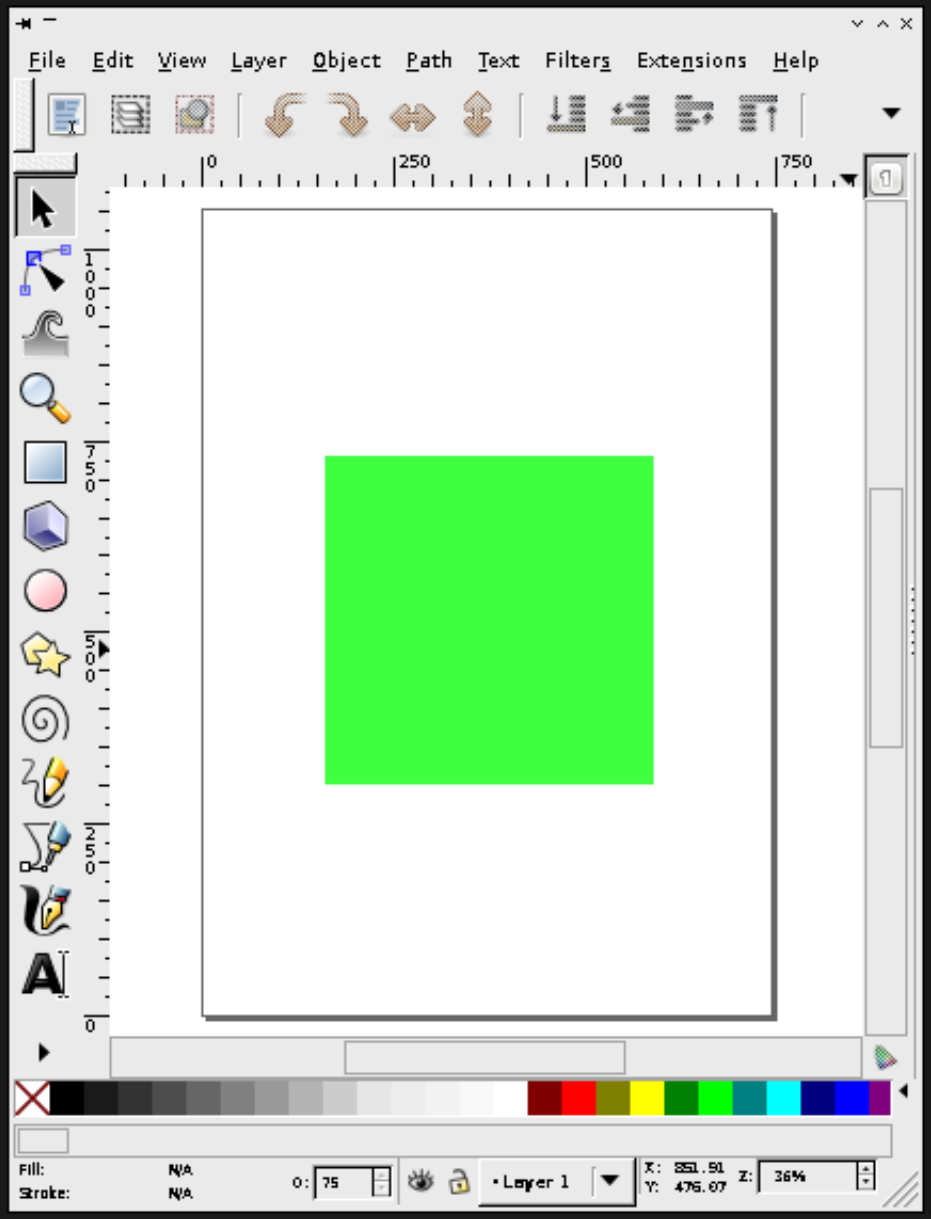
ASCII text is easily read and edited.

```
#  substitute  "foo"  with  "bar"
sed  -i  's/foo/bar/g'  input.txt
```


File Edit View Layer Object Path Text Filters Extensions Help
foo
Fill: N/A
Stroke: N/A
O: 75
Layer 1
X: 854.70
Y: 250.96
Z: 36%


File Edit View Layer Object Path Text Filters Extensions Help
bar
> sed -i 's/foo/bar/g' inkscape+sed.svg
>


File Edit View Layer Object Path Text Filters Extensions Help
0
250
500
750
Fill: N/A
Stroke: N/A
O: 75
Layer 1
X: 799.62
Y: 938.22
Z: 36%
> sed -i 's/foo/bar/g' inkscape+sed.svg
>




```
> sed -i 's/foo/bar/g' inkscape+sed.svg
> sed -i 's/ff0000/00ff00/g' inkscape+sed.svg
>
```

sed handy one-liners

http://sed.sourceforge.net/sed1line.txt

Automate everything.

```
BACKGROUND=`ls $DIR/*.pdf | head -1`

for PDF in `ls $DIR/*.pdf | grep -v $BACKGROUND`
 do
        pdftk $PDF background $BACKGROUND output $OUTPUT
        BACKGROUND=$OUTPUT
done
```

```
BACKGROUND=`ls $DIR/*.pdf | head -1`

for PDF in `ls $DIR/*.pdf | grep -v $BACKGROUND`
 do
        pdftk $PDF background $BACKGROUND output $OUTPUT

        pdf2ps $OUTPUT ${OUTPUT%%.*}.ps
        ps2pdf ${OUTPUT%%.*}.ps $OUTPUT
        rm ${OUTPUT%%.*}.ps

        BACKGROUND=$OUTPUT
done
```

2014

human readable source files

+

accessible render engine

ASCII source files

commandline support

Showstopper?

binary file formats
lacking commandline support
captive user interfaces

Future?

human readable source files

+

accessible render engine

Distrust all claims for one true way

Thanks:

Martin Rumori/Frank Barknecht

goto10

servus.at

LGRU

Constant

Free Software Developers

You