Extending prerendered-interface voting software to support accessibility and other ballot features

Ka-Ping Yee
UC Berkeley
ping@zesty.ca

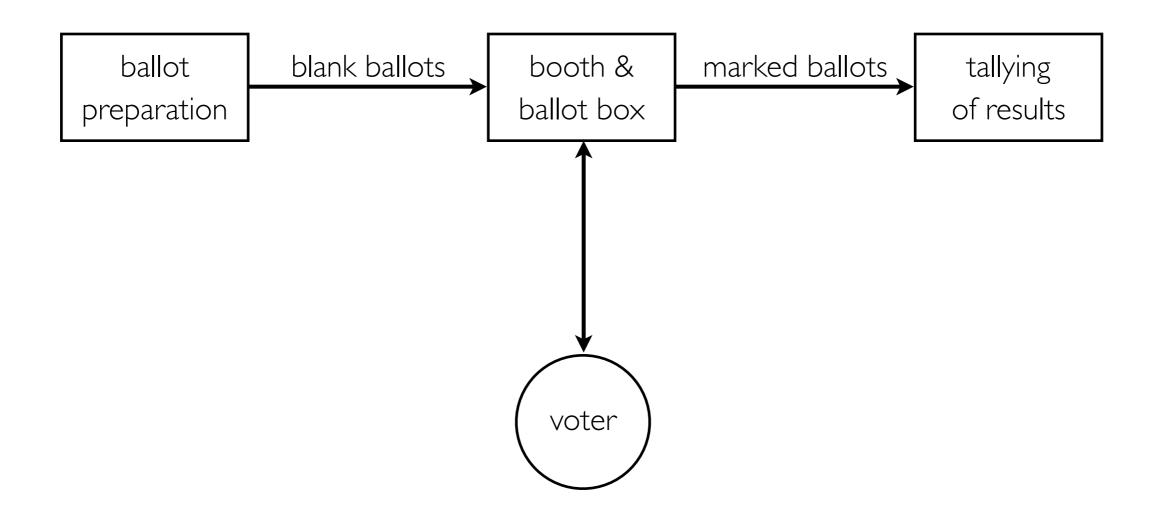
I. The prerendered-interface paradigm

2. Pvote demonstration

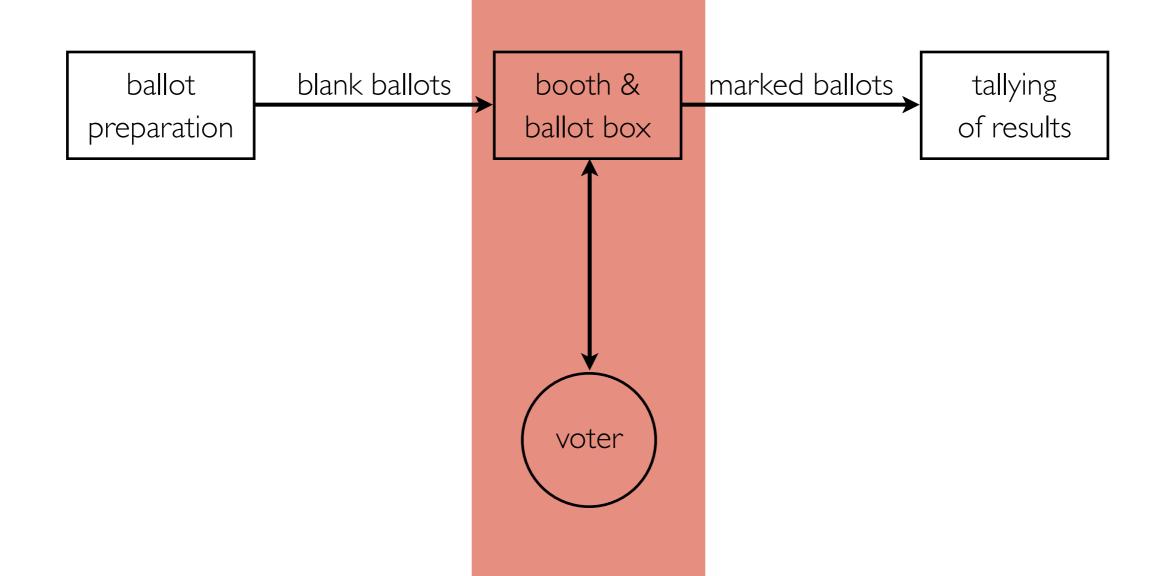
3. Design

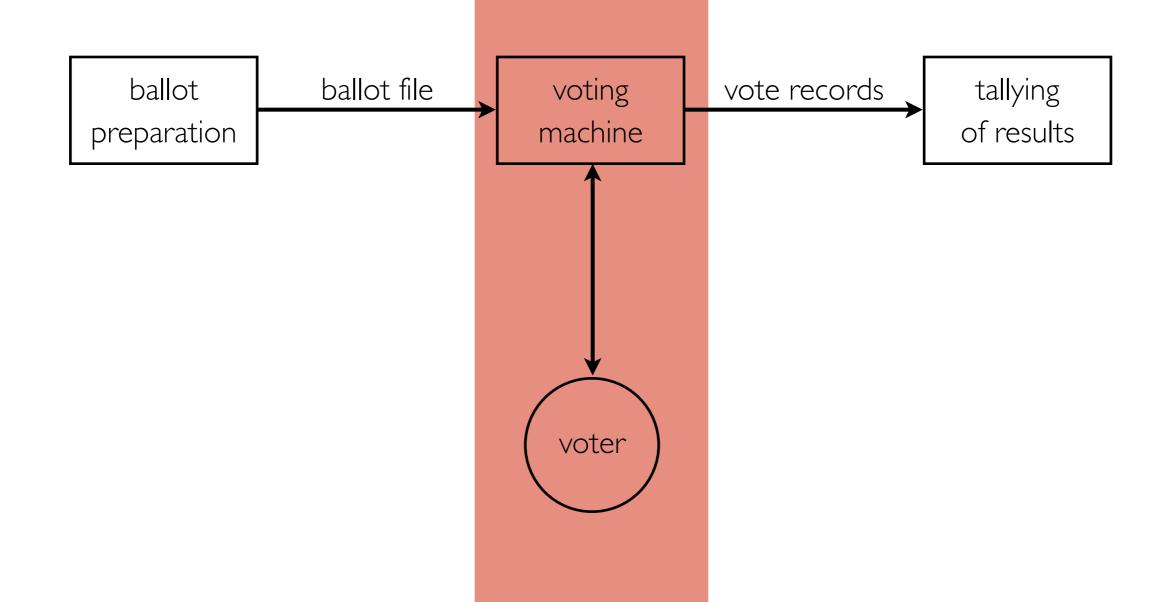
4. Complexity and assurance

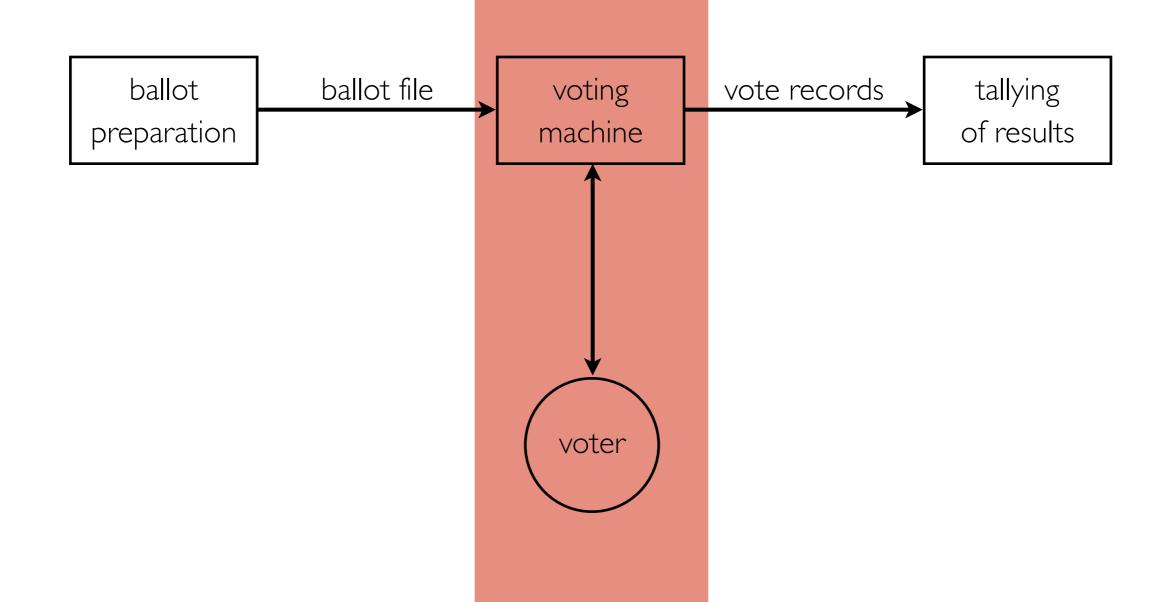
voting on paper



voting on paper



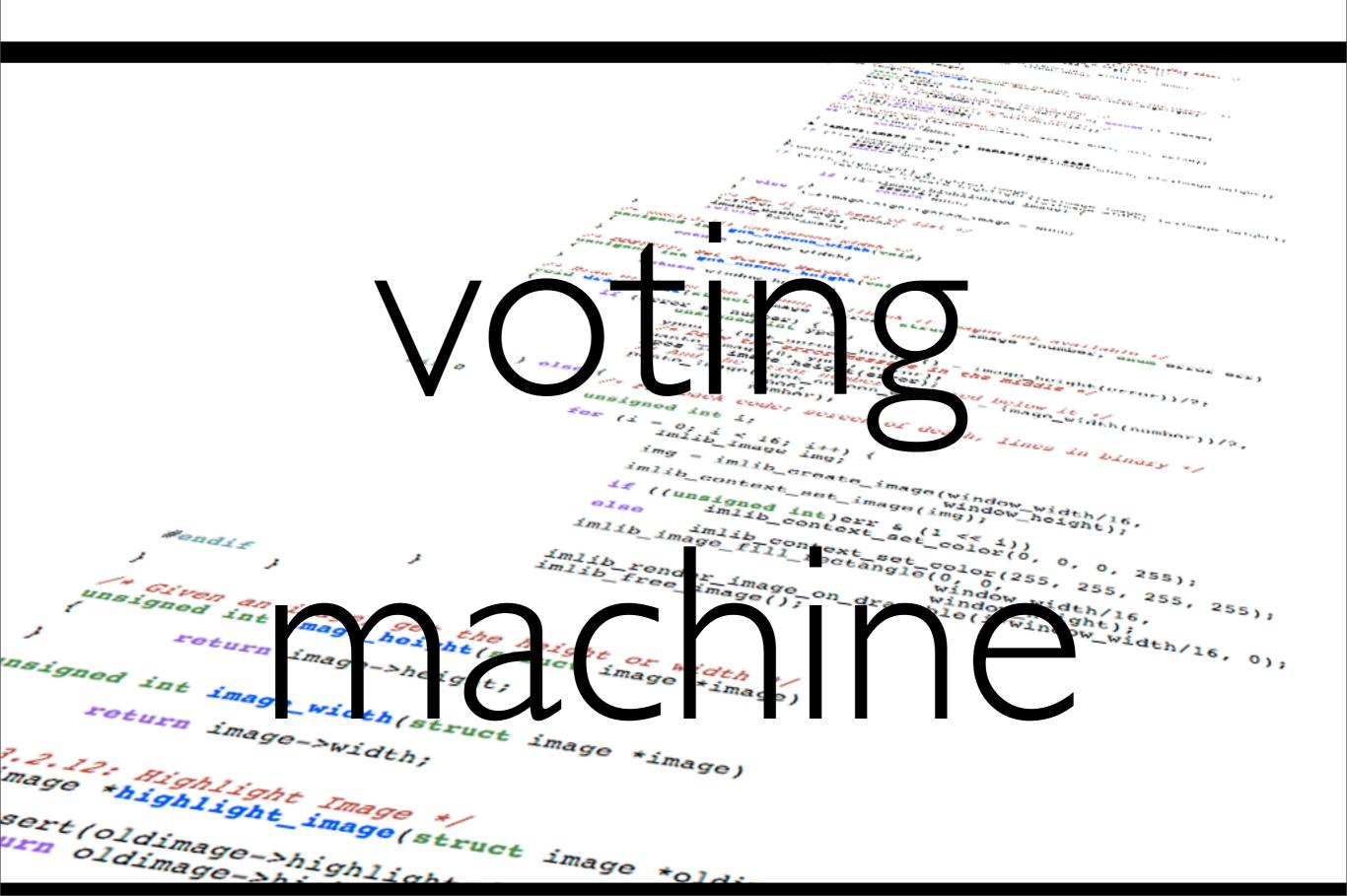




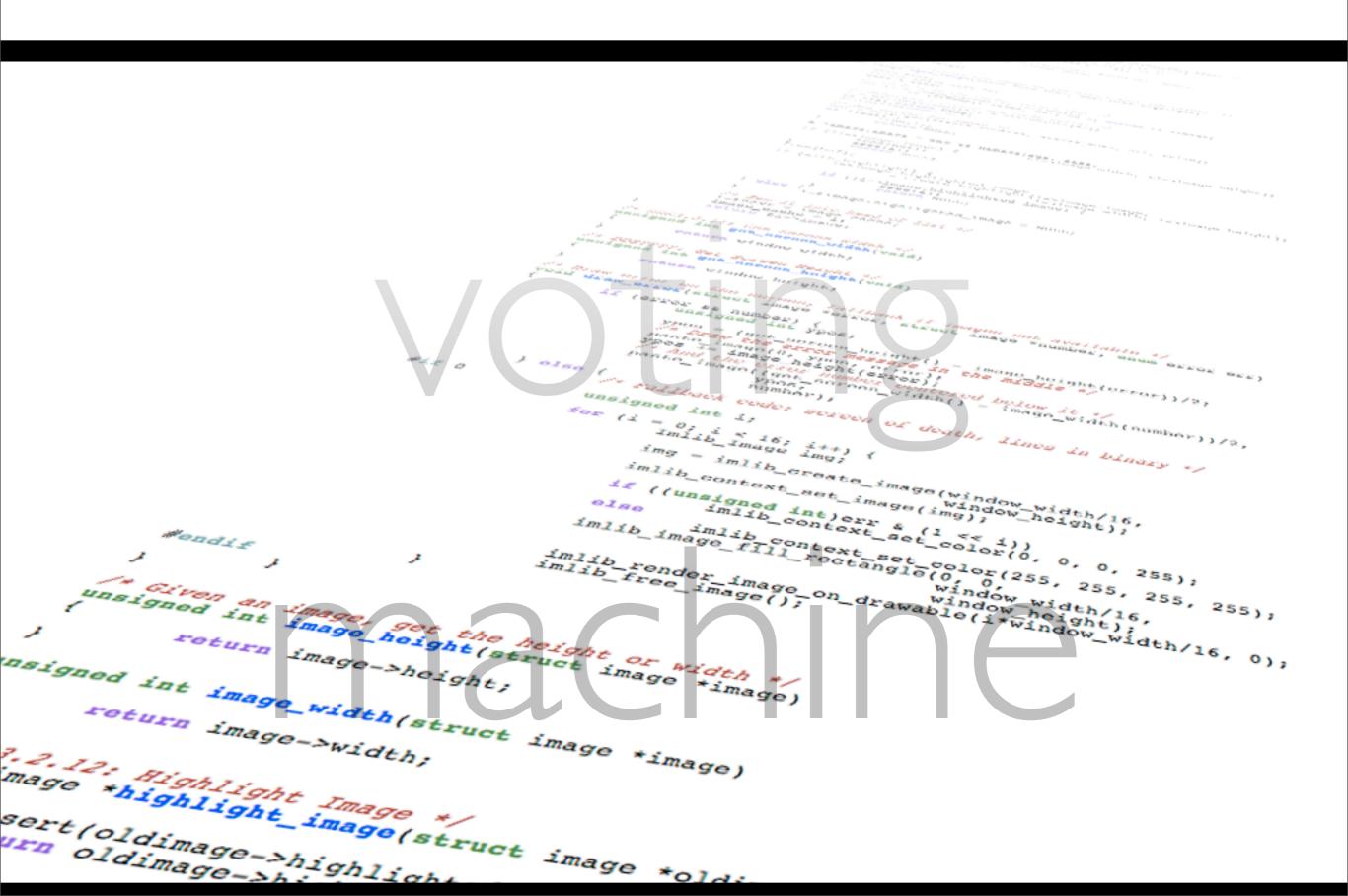
Voting machine

Voting machine

Voting machine



```
66 000 lines of C++
       Diebold AccuVote-TSx:
                                                     22 000 lines of C++
       Hart eSlate:
       Sequoia Edge:
                                                    124 000 lines of C
                                                                      wideh ( maga wideh (number))/2,
                                                               sessed of degen, Lines in binary "
                                                    Imlib Image itt)
                                                  img = imlib_create_image(window_width/16, imlib_context_set_image(img); window_hoight);
                                                 imlib_context_met_image(img);
                                               imlib_context_set_color(0, 0, 0, 255);
                                          imlib_imlib_context_set_color(255, 255, 255);
imlib_render_image on window width/16, 0);
imlib_free_image();
imlib_free_image();
imlib_free_image();
            Pondif
                      image hoight (struct
             roturn image->height;
                                        height or width */
truct image *image)
nsigned int image width(struct image *image)
7.2.12: Bighlight Image */
mage *highlight_image(struct image *old:
```



widget vote storage language look & feel handling logic graphics text layout device drivers & rendering drawing ght (s nsign) ; selection & interaction logic font data Mage *highlight Image */
image(struct image *old:

language handling

widget look & feel vote storage logic

text layout & rendering

graphics drawing device drivers

O);

font data

selection & interaction logic

widget vote storage language look & feel logic handling text layout graphics device & rendering drawing drivers selection & selection & font data interaction interaction

language handling widget look & feel vote storage logic

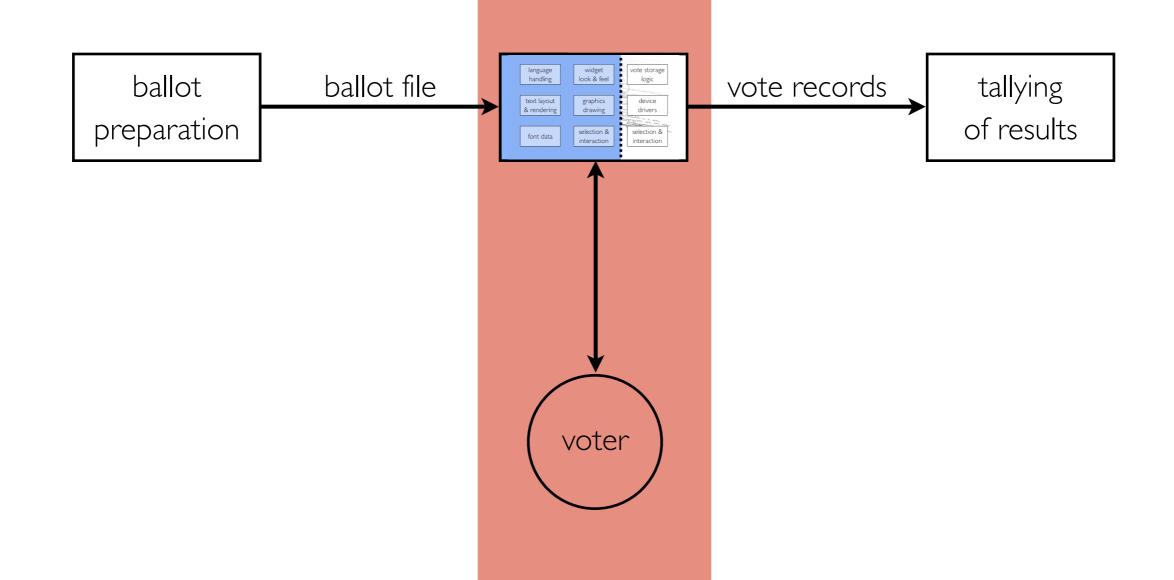
text layout & rendering

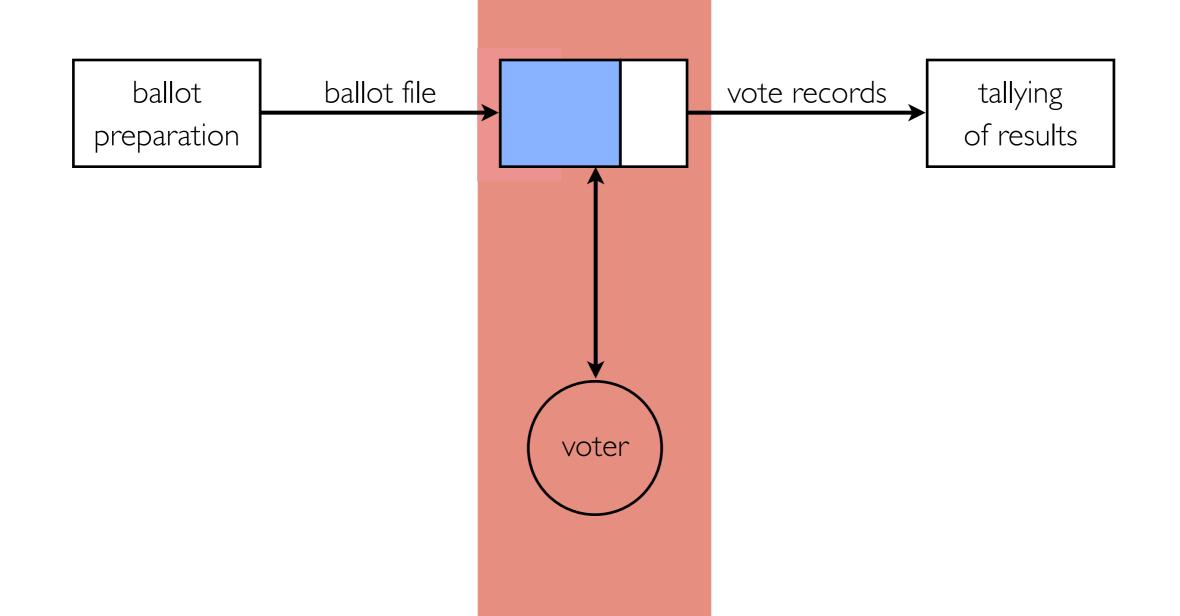
graphics drawing device drivers

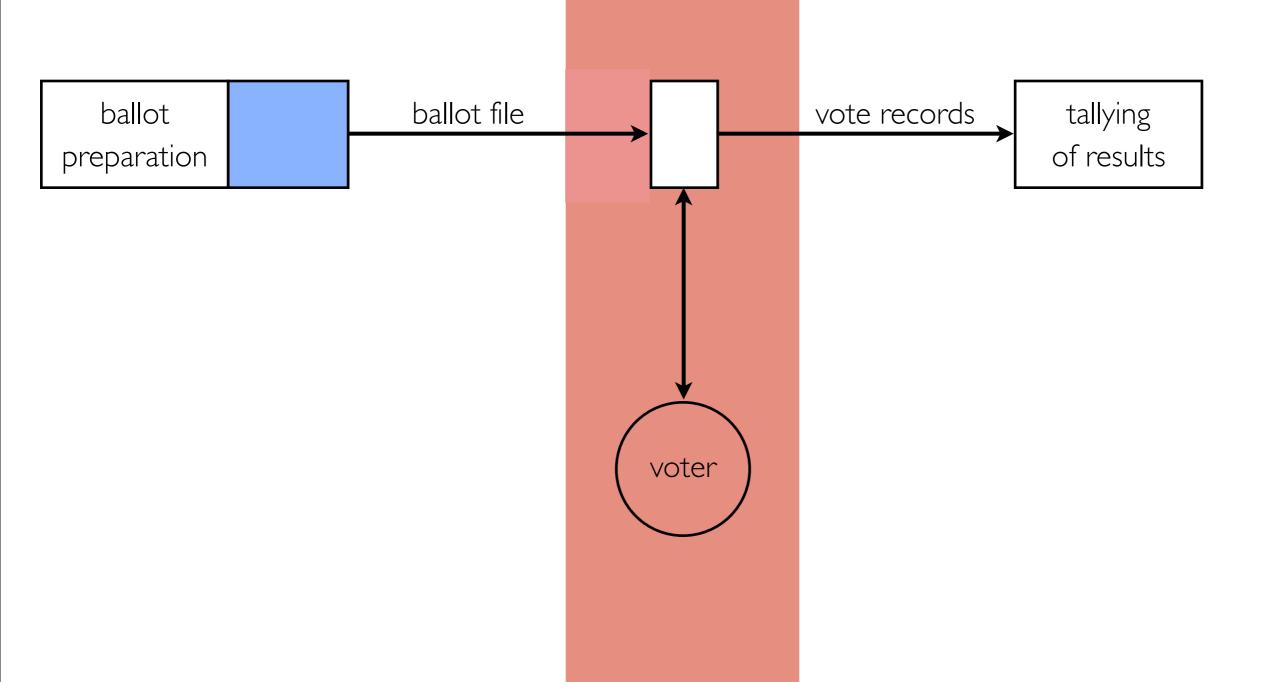
font data

selection & interaction

selection & interaction







Diebold AccuVote-TSx: 66 000 lines of C++

Hart eSlate: 22 000 lines of C++

Sequoia Edge: 124 000 lines of C

Diebold AccuVote-TSx: 66 000 lines of C++

Hart eSlate: 22 000 lines of C++

Sequoia Edge: 124 000 lines of C

Pvote: 460 lines of Python

multimodal interface:

```
output { audio only video only synchronized audio + video }

touchscreen buttons / keys assistive devices
```

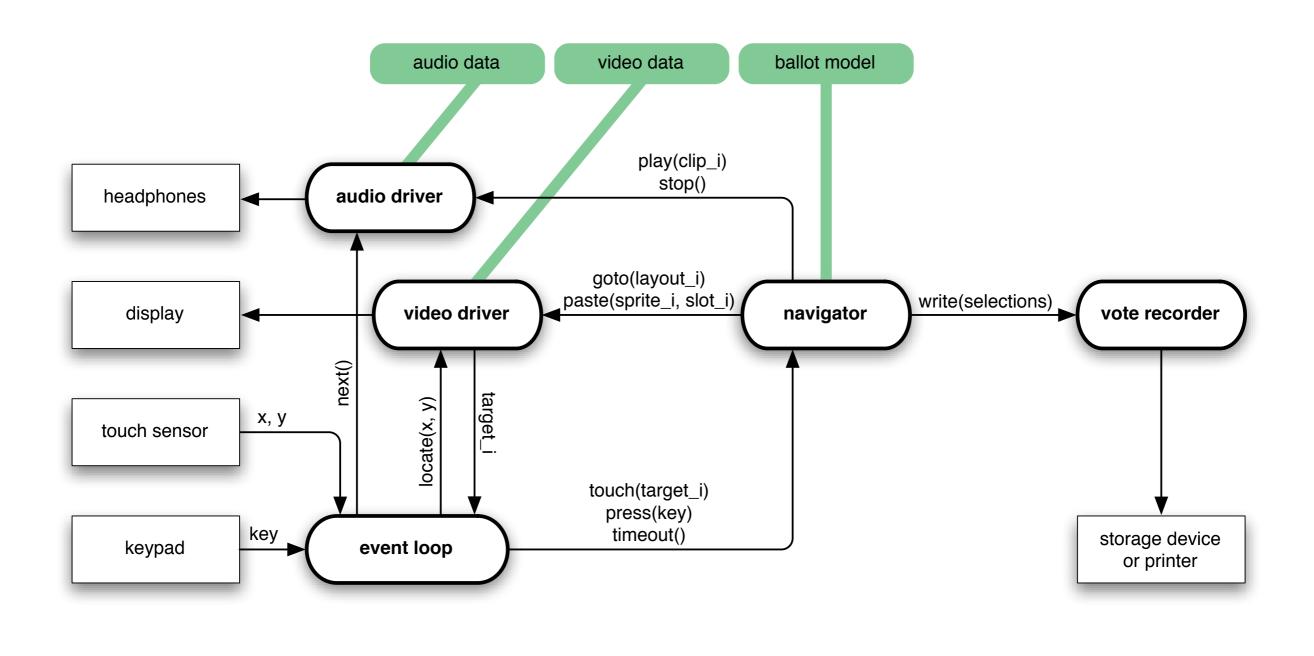
vote for one of n

373 ballots

vote for one of <i>n</i>	373
vote for up to k of n	195
vote yes or no	251
ranked choice (up to 3)	7
write-in candidate	318
straight-party vote	60
cross-endorsed candidates	8
multi-party primary	5
party logo images	21
ideographic languages (CJK)	48
multiple languages	53

vote for one of n	373
vote for up to k of n	195
vote yes or no	251
ranked choice (up to 3)	7
write-in candidate	318
straight-party vote	60
cross-endorsed candidates	8
multi-party primary	5
party logo images	21
ideographic languages (CJK)	48
multiple languages	53

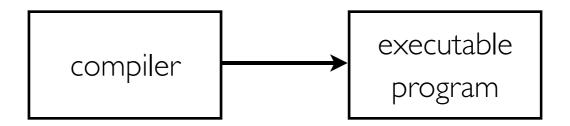
all supported by Pvote



one-way data flow ballot definition software module hardware device

program

executable program

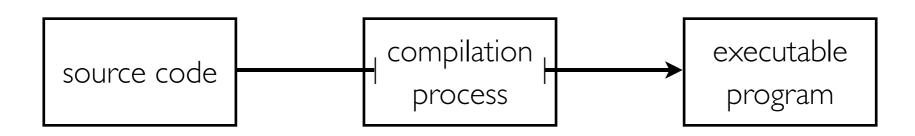


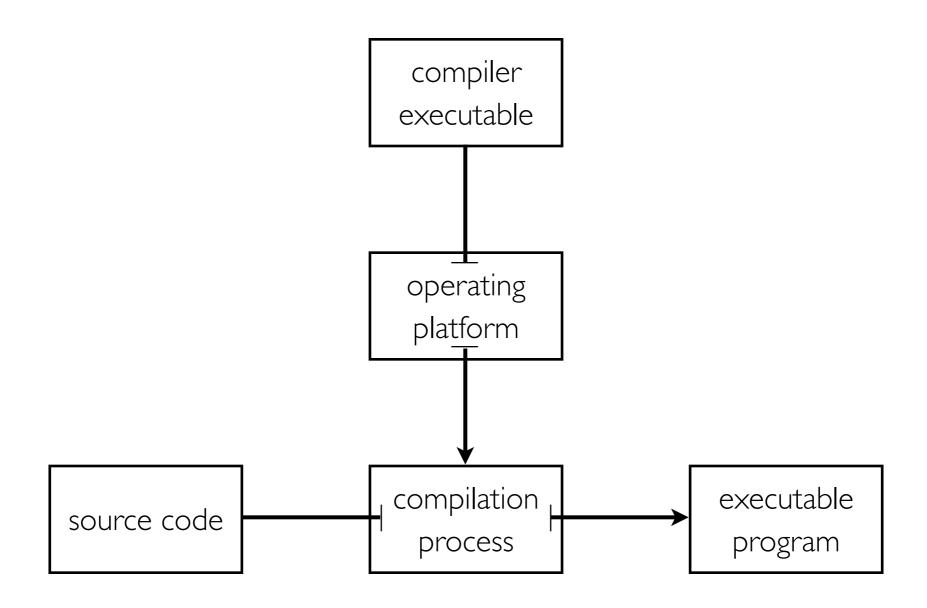
source code

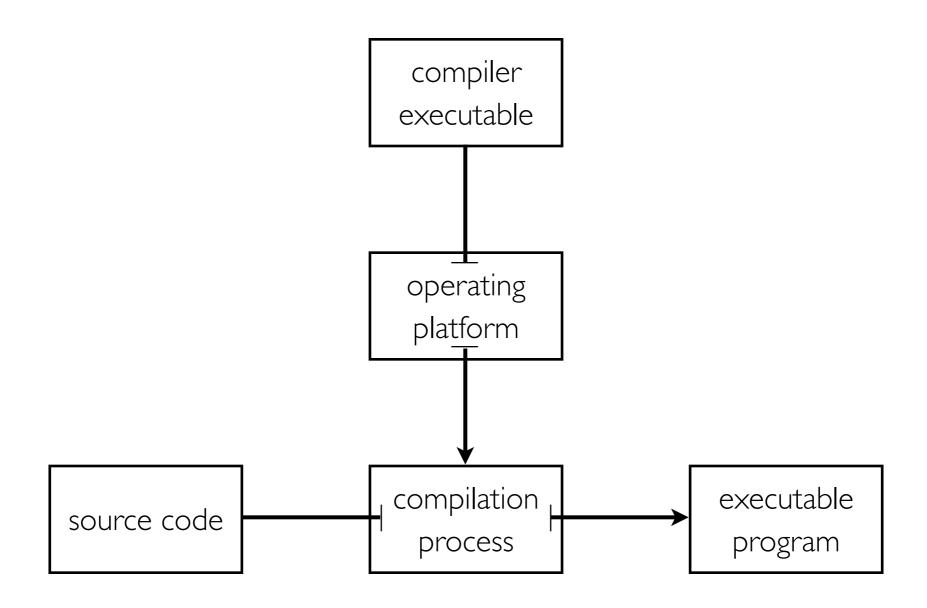
compiler

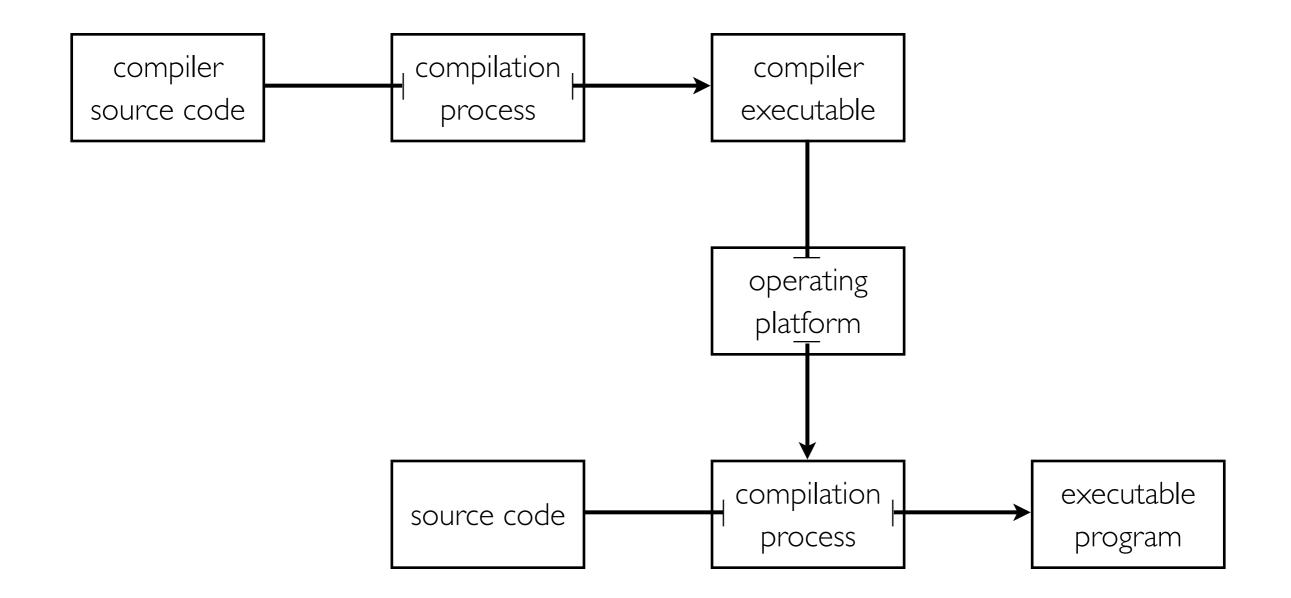
executable program

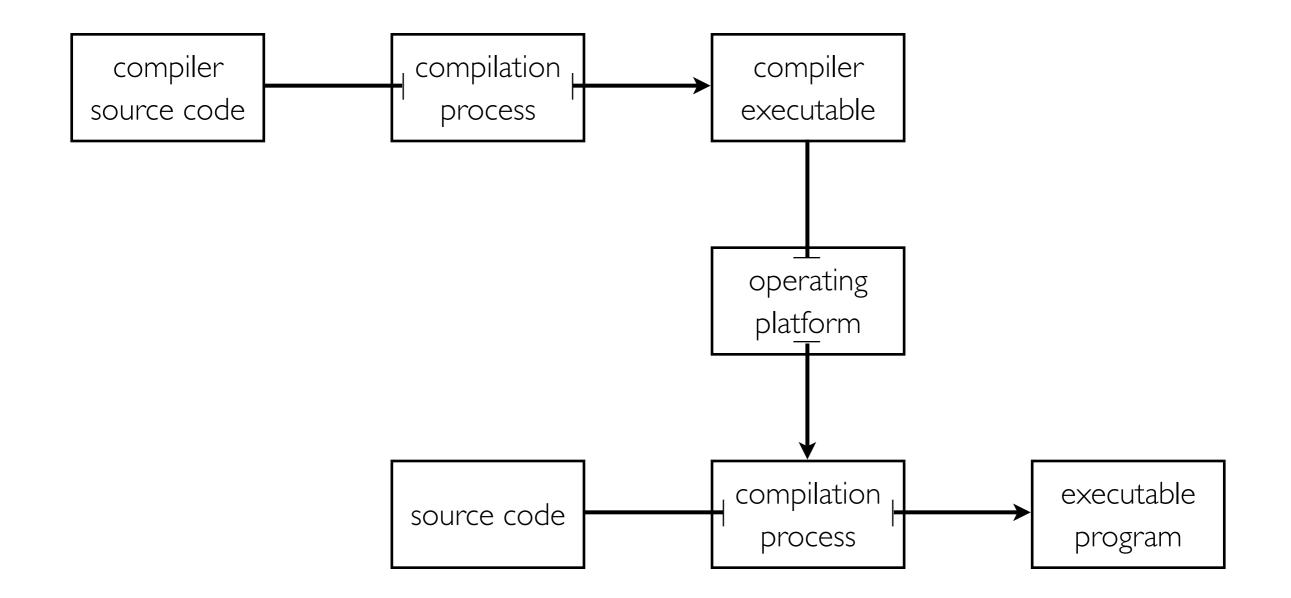


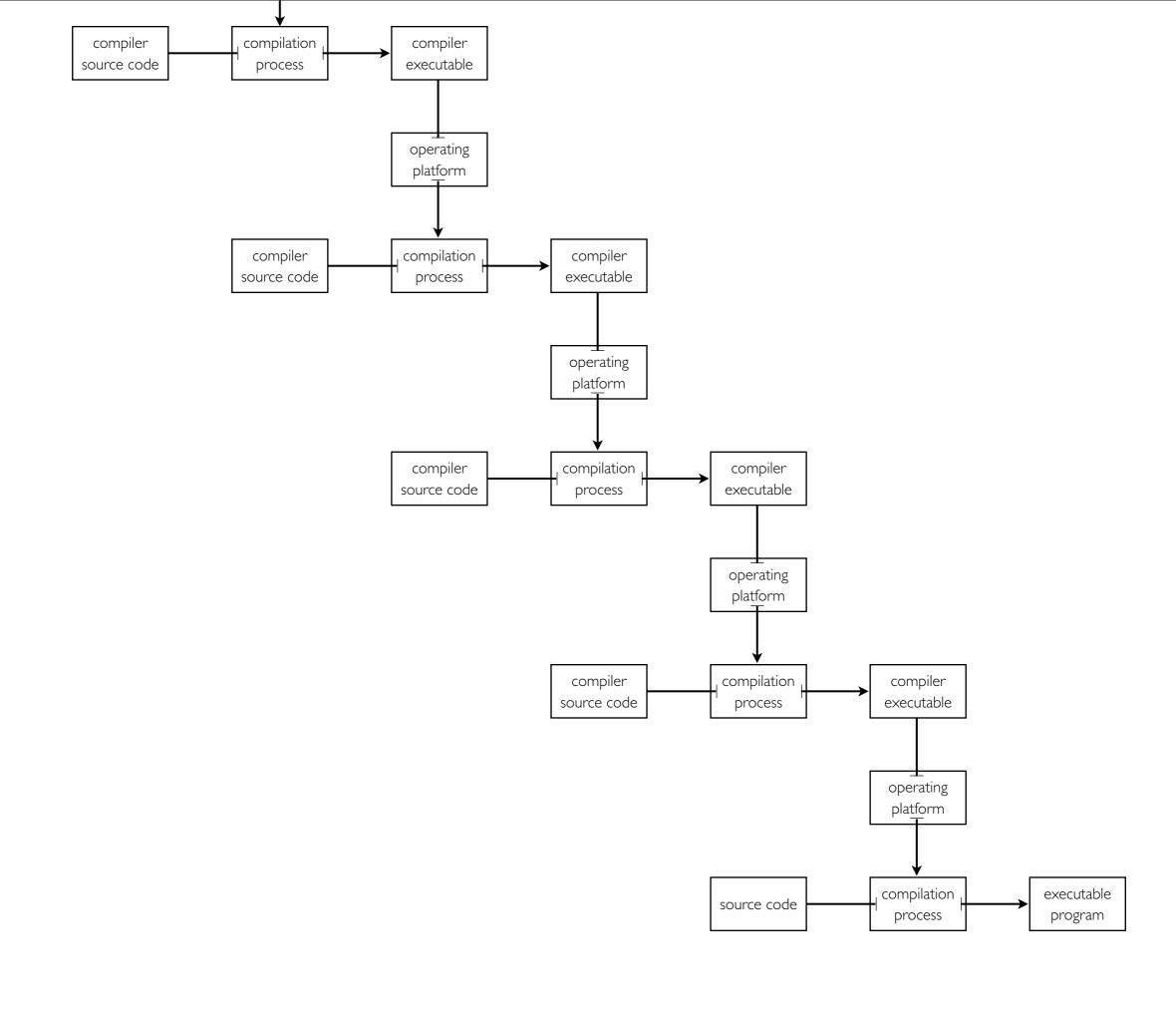












voting user experience

