"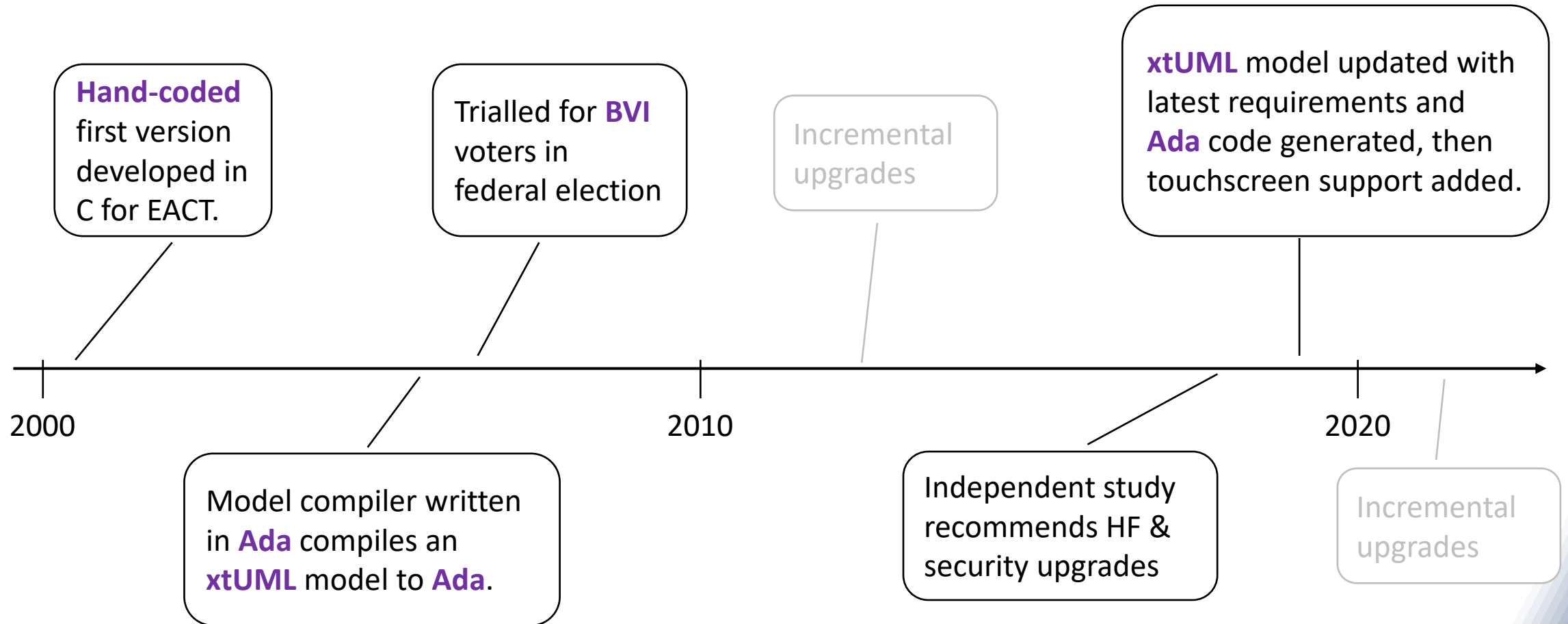I consider it completely unimportant who in the party will vote, or how; but what is ***extraordinarily important*** is this — ***who will count the votes, and how.***"

Boris Bazhanov, *The Memoirs of Stalin's Former Secretary* (1992)

# eVACS® timeline

**Hand-coded** first version developed in C for EACT.

Trialled for **BVI** voters in federal election

Incremental upgrades

**xtUML** model updated with latest requirements and **Ada** code generated, then touchscreen support added.

2000

2010

2020

Model compiler written in **Ada** compiles an **xtUML** model to **Ada**.

Independent study recommends HF & security upgrades

Incremental upgrades

DedicatedSystems

"The industry of software should make efforts to understand the issues related to **Human Factors** within data collection and the **entire process of development**"
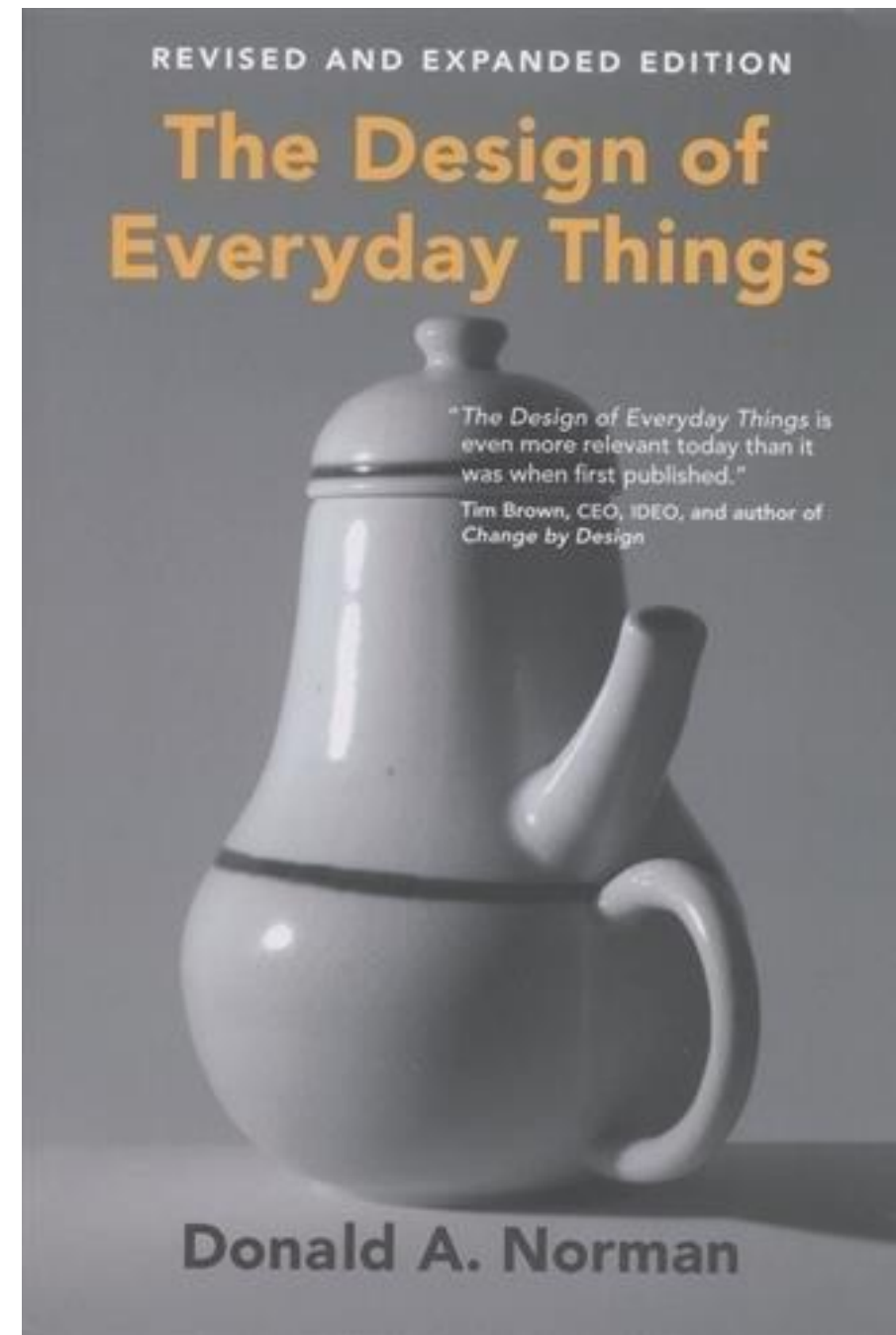
*– ERAU.*

Synthesize **knowledge**
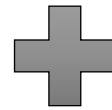
Make functions **simple**

Make functions **visible**

Develop **consistent models**

Leverage **constraints**

Assume **errors**



REVISED AND EXPANDED EDITION

The Design of Everyday Things

"The Design of Everyday Things is even more relevant today than it was when first published."
Tim Brown, CEO, IDEO, and author of Change by Design

Donald A. Norman

# eVACS 2001 essential requirements

There is no Internet Connection

**+**

Open source

Limited interfaces

Limited access

One voter, one vote

Accurate counts

Recoverability

DedicatedSystems
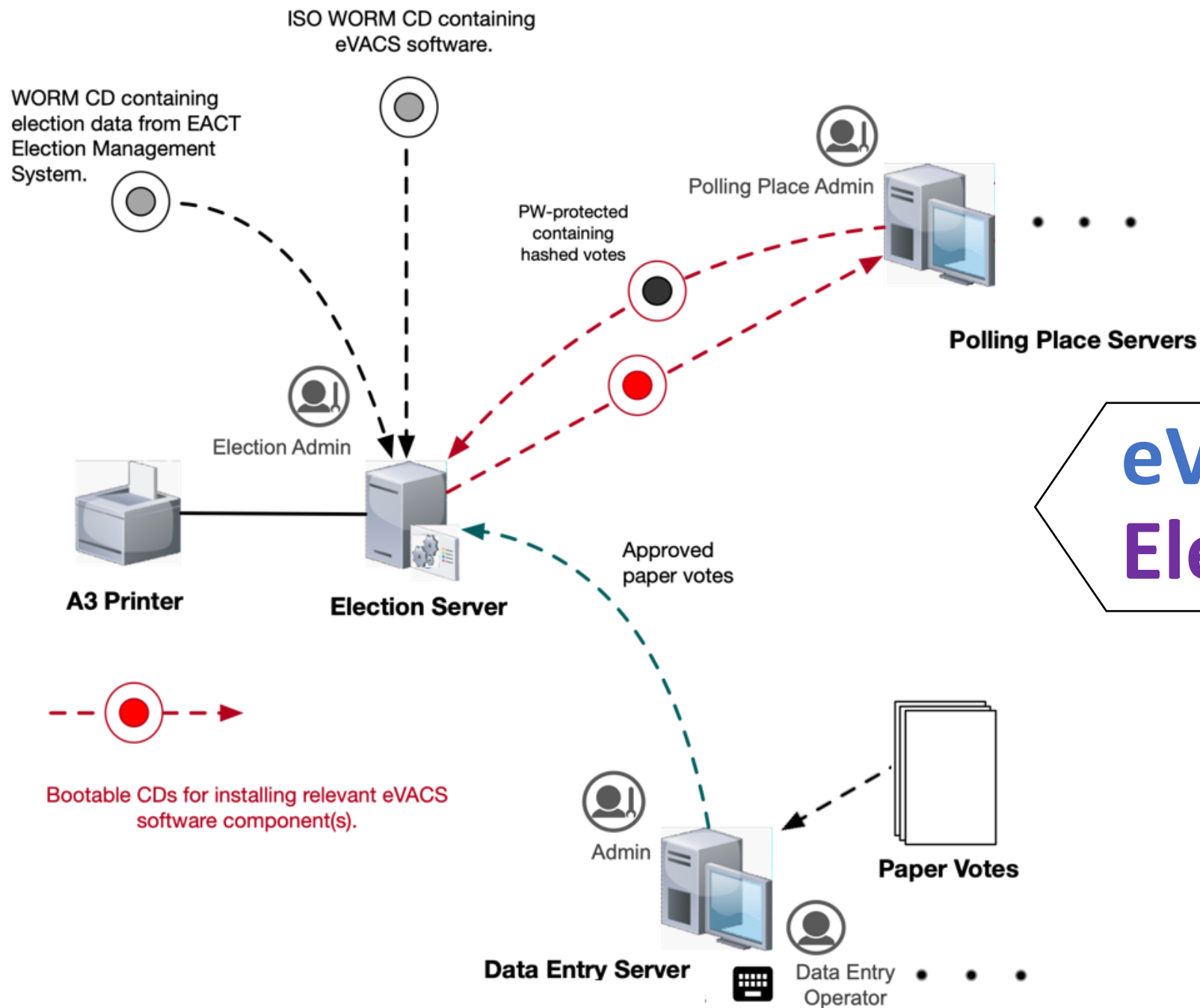
eVACS® in 2001: Election Office

Each Polling Place Sever is created from a specific Boot CD.

Polling Place Admin

Elector

Elector

BVI Elector

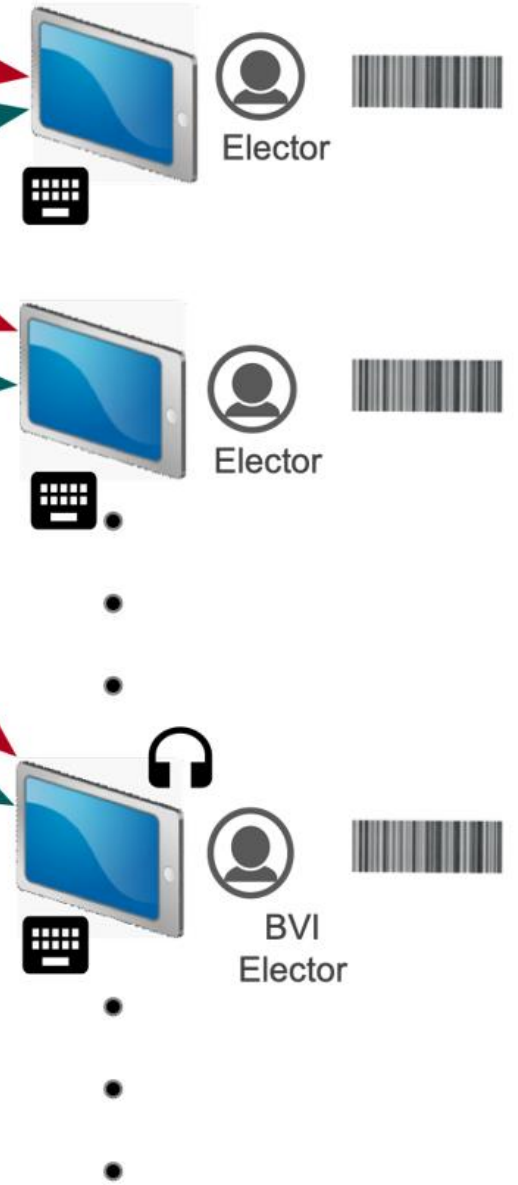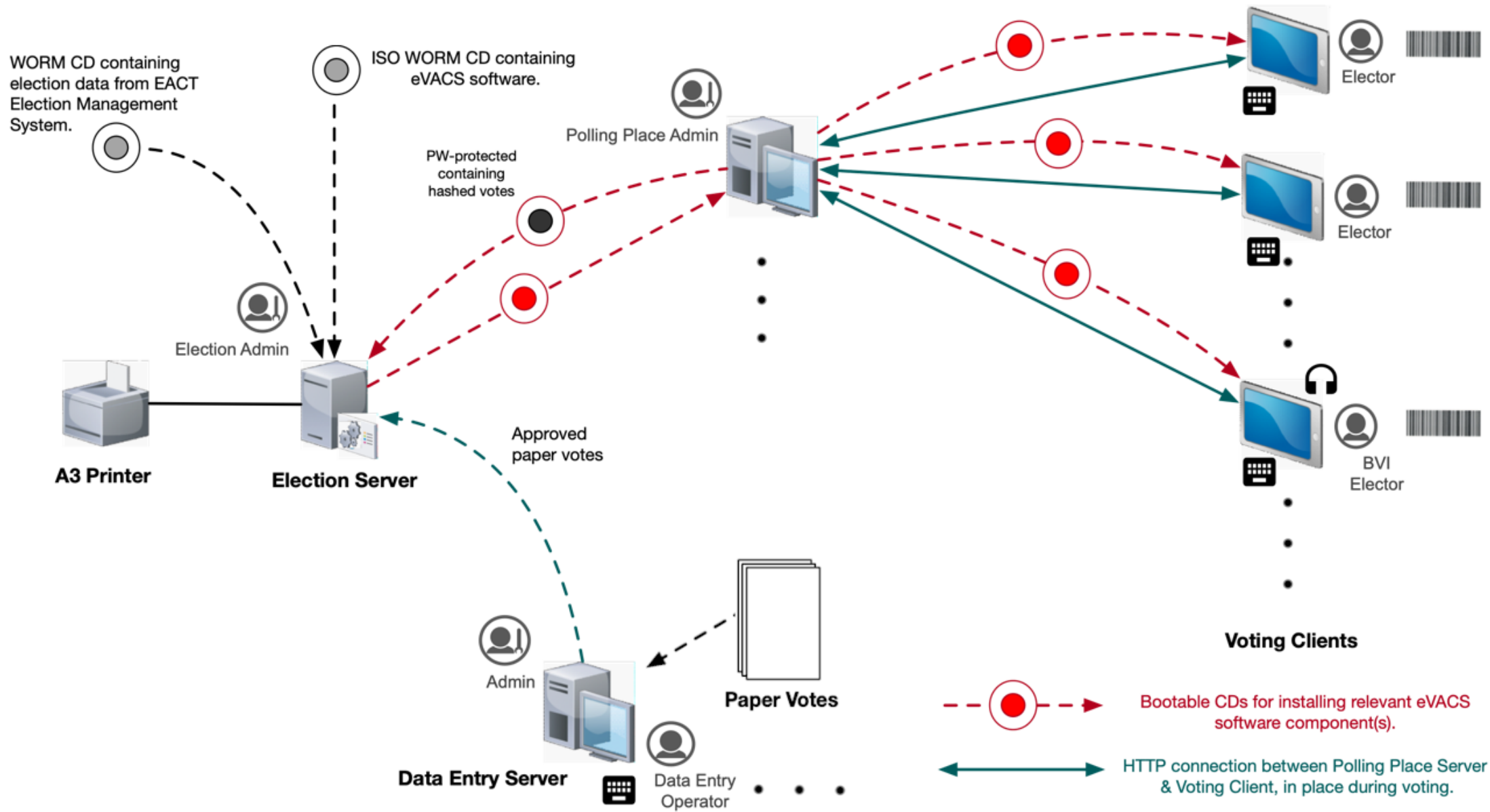**eVACS® in 2001: Polling Place**

Bootable CDs for installing relevant eVACS software component(s).

HTTP connection between Polling Place Server & Voting Client, in place during voting.

**Voting Clients**

WORM CD containing election data from EACT Election Management System.

ISO WORM CD containing eVACS software.

PW-protected containing hashed votes

Polling Place Admin

Election Admin

A3 Printer

Election Server

Approved paper votes

Admin

Paper Votes

Data Entry Server

Data Entry Operator

Elector

Elector

BVI Elector

Voting Clients

Bootable CDs for installing relevant eVACS software component(s).

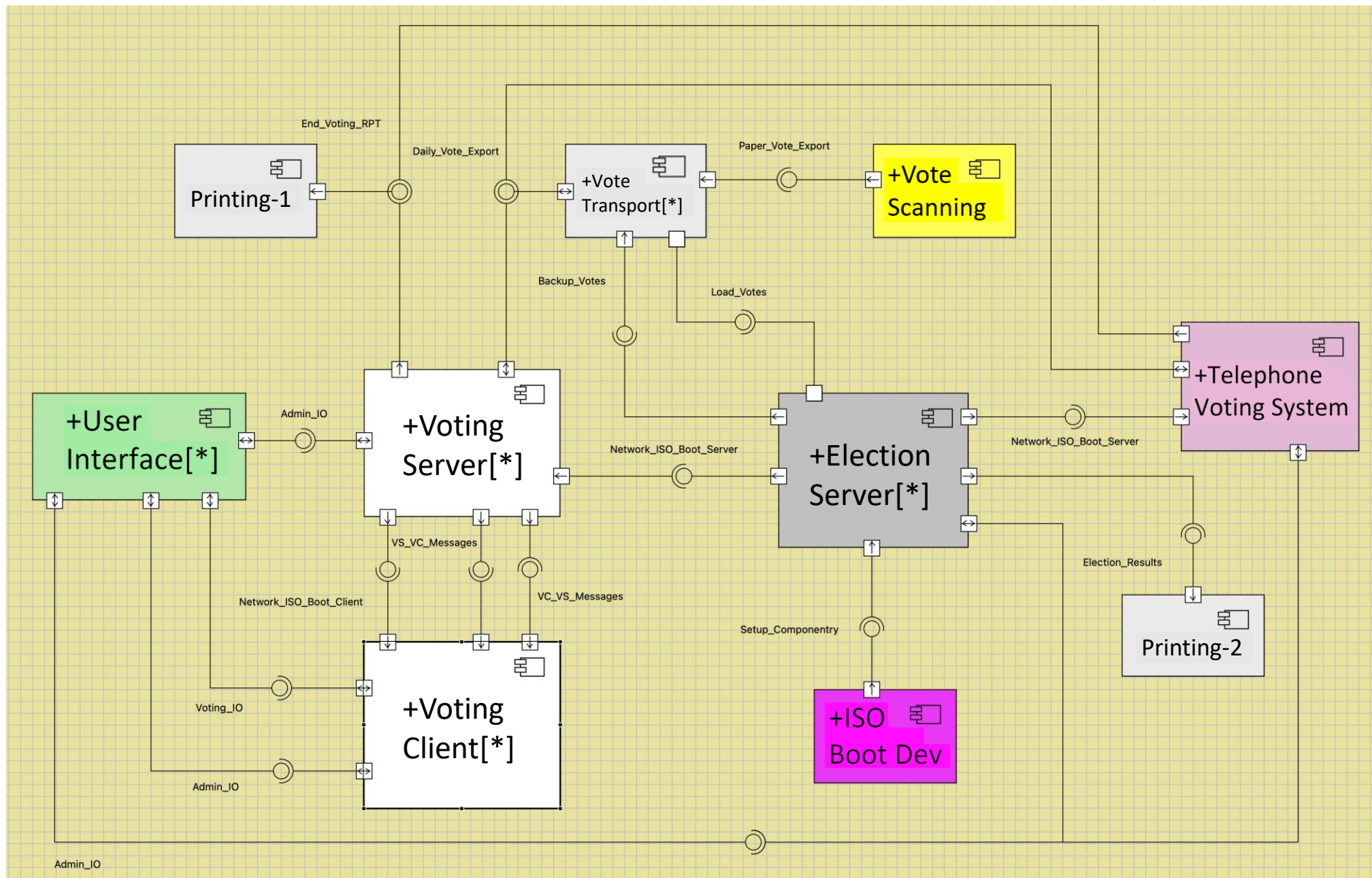HTTP connection between Polling Place Server & Voting Client, in place during voting.

DedicatedSystems

# eVACS® in 2020/24: The updated human view
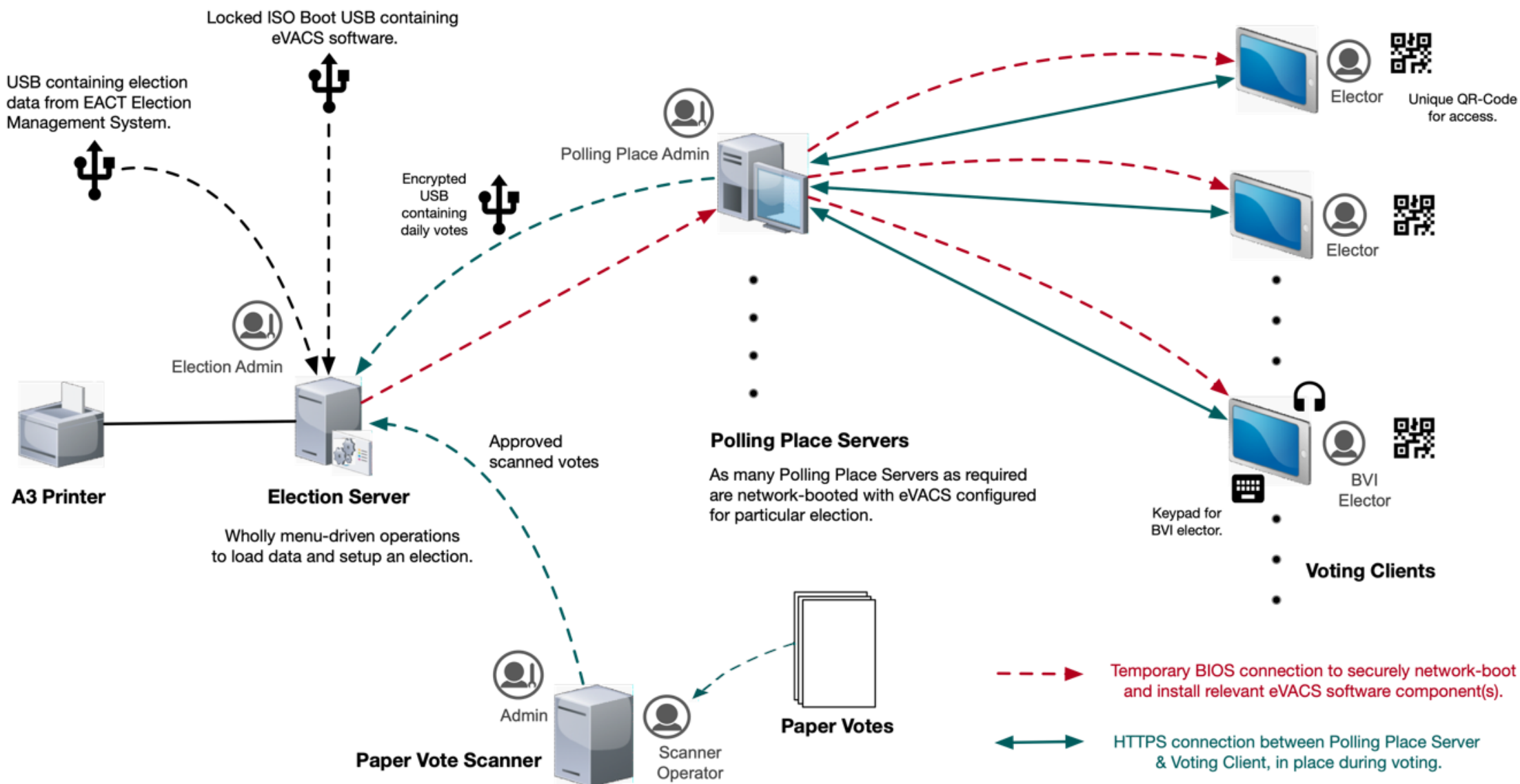
eVACS® 2020: The updated architectural view

eVACS® in 2020: What the Voter sees – 2

eVACS® in 2020: What the Voter sees – 3a

Elections ACT

For help, raise your hand and press
**HIDE MY VOTE**

**Review your choices**
To change your choices touch **GO BACK**
OR
To confirm your choices **scan e-voting card** to cast your vote

Your choices in preference order are

1 Candidate name
A **PARTY NAME**

2 Candidate name
D **PARTY NAME**

3 Candidate name
A **PARTY NAME**

4 Candidate name
C **PARTY NAME**

5 . Candidate name
E **PARTY NAME**

Privacy of vote

Progress of vote

**GO BACK**    ✓ ✓ 3 4

DedicatedSystems

**Elections** ACT

**You have not selected any candidates.**
**Do you want to cast a blank informal vote?**

No, I want to return to the ballot paper  GO BACK

Yes, I want to  CAST AN INFORMAL VOTE

GO BACK  ✓ ✓ 3 4

DedicatedSystems

**Attributes**

1 is included in

+Contest
{7,VCM_CON}

Election_Name:string
Contest_Name:string {I2}
Contest_ID:integer {I}
Ballot_Type:ballot_type
Rotation_Type:rotation_type
Min_Candidates:integer
Max_Candidates:integer
Last_Preference_Implied:boolean
Number_Of_Positions:integer

R14

1 rotates candidate choices for

DedicatedSystems

**eVACS in 2020: The control model view**

+LA_Ballot
{114,LAB}

current_state:state<State_Model>
voting_method:voting_method
current_state:state<State_Model>

process_input(input:string):void
is_formal():boolean
play_intro_audio():void
display_screen():void
play_informal_audio():void
add_confirmation_audio():void
add_intro_audio():void
add_informal_audio():void
set_confirmation_screen_help_audio():void
add_audio_for_current_selection():void
queue_preference_loop(first:integer, chunk_size:integer):void
play_audio_for_current_selection():void

**Operations**

DedicatedSystems

## 12. Add Preference

```
entry/
select any selectable_item from instances of VCM_SI where selected.IsSelected == true;
select one choice related by selectable_item -> VCM_CC[R10];
select one group related by selectable_item -> VCM_GRP[R10];
group_is_selected = not empty group;
if empty group
  select one group related by choice -> VCM_GRP[R2];
end if;
select one contest related by group -> VCM_CON[R1];
select one vote related by contest -> VCM_BAL[R4] -> VCM_VOT[R11];
select many below_the_line_preferences related by vote -> VCM_PRF[R12];

//voting below the line, so no adding preferences to groups
if group_is_selected
  generate VCC_SEN1:'Wait For Input' to self;
  return;
end if;

//Check that there isn't already a preference for this choice
select one preference related by choice ->VCM_PRF[R12];
if not empty preference
  generate VCC_SEN1:'Wait For Input' to self;
  return;
end if;

create object instance preference of VCM_PRF;
preference.PreferenceNumber = cardinality below_the_line_preferences + 1;
relate choice to vote across R12 using preference;
```

```ada
with Gtk.Button; use Gtk.Button;
with eVACS.Types; use eVACS.Types;

package Voting_Client_Interface is

   procedure Initialise_Voting_Client;
   procedure Barcode_Swipe (Input : in String);
   procedure Set_Language (Language_Id : Natural);
   procedure Shutdown;
   procedure Add_Touch_Pos (party_id : Natural;  candidate_id : Natural;  cand_id : Natural);
   procedure Undo_Last_Preference;
   procedure Report_Error (error : error_code_Type; message : String);
   function Get_Last_Error return error_code_Type;
   procedure On_Completed;
   procedure On_Run_ScanQR;
   procedure On_Next_Clicked (Button : access Gtk_Button_Record'Class);
   procedure On_Cast_Informal_Clicked (Button : access Gtk_Button_Record'Class);
   procedure On_Hide_Vote_Clicked (Button : access Gtk_Button_Record'Class);
   procedure On_Start_Again_Clicked (Button : access Gtk_Button_Record'Class);
   procedure On_Clear_Choices_Clicked (Button : access Gtk_Button_Record'Class);
   procedure On_Back_Clicked (Button : access Gtk_Button_Record'Class);
   procedure On_Continue_Voting_Clicked (Button : access Gtk_Button_Record'Class);
   procedure On_Vote_Release_Clicked (Button : access Gtk_Button_Record'Class);
   procedure On_Vote_Release_Other_Clicked (Button : access Gtk_Button_Record'Class);
   procedure On_Change_Language_Clicked (Button : access Gtk_Button_Record'Class);
   function Is_BVI return Boolean;
   procedure Input_Event (Input :  String);
   function Vote_Keypress (Keypress : String) return String;
   function Check_Response (Strm : String) return Boolean;
end Voting_Client_Interface;
```

# eVACS 2020: An Ada Case Statement

```ada
procedure Process_Key_Event (Self : not null access VC_Main_Voting_Screen_Record; Keypress : String) is
Results : JSON_Value;
  begin
    case Keypress (Keypress'First) is
      when '0' => Play_Audio (Self, Audio_Help_Zero);
      when '1' => VCI.On_Clear_Choices_Clicked (null);
      when '2' | '4' | '5' | '6' | '8' | '*' | '#' => Results := Read (VCI.Vote_Keypress (Keypress));
        if VCI.Check_Response (Results.Write) then
          if not Results.Get (Field_Results).Has_Field (Field_Cursor_Pos) then
            raise Server_Param_Error with "cursor_pos field not found";
          end if;
          if Results.Get (Field_Results).Get (Field_Cursor_Pos).Get (Field_Action) = Continue'Image then
            Update_Preference (Self, Results.Write);
            Play_Audio (Self, Audio_Continue);
          elsif Results.Get (Field_Results).Get (Field_Cursor_Pos).Get (Field_Action) = Finish'Image then
            VCI.On_Next_Clicked (null);
          end if;
        end if;
      when '3' => VCI.On_Hide_Vote_Clicked (null);
      when '7' | '9' => null;
      when others => null;
    end case;
  exception
    when Occurrence: others =>
      if VCI.Get_Last_Error = ERR_OK then
        VCI.Report_Error (ERR_INTERNAL, Exception_Name (Occurrence) & ':' & Exception_Message (Occurrence));
        null;
      end if;
  end Process_Key_Event;
```

# Key Takeaways

## Human factors are ubiquitous

## Affordance & Feedback are key