

Blaise on touch-screen tablets: The ELIPSS example

Mathieu Olivier, Alexandre Chevallier

▶ **To cite this version:**

Mathieu Olivier, Alexandre Chevallier. Blaise on touch-screen tablets: The ELIPSS example. IBUC 2016 17th International Blaise Users Conference, International Blaise User Group, Oct 2016, La Haye, Netherlands. hal-03571058

HAL Id: hal-03571058

https://sciencespo.hal.science/hal-03571058

Submitted on 13 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - ShareAlike| 4.0 International License

Blaise on Touch-Screen Tablets: the ELIPSS example

Alexandre Chevallier
Sciences Po, Center for socio-political data (CDSP), CNRS, Paris, France

Mathieu Olivier
Sciences Po, Center for socio-political data (CDSP), CNRS, Paris, France

IBUC 2016

Wednesday, 5th of October, The Hague

ELIPSS in few words

- *Etude Longitudinale par Internet Pour les Sciences Sociales*
- Probability-based online panel
- Pilot study 2012-2016: 1 039 individuals
- Monthly questionnaires designed by researchers
- Internet access given to **each** panel member

7-inch touchscreen tablet
& 3G subscription



Tablets & Internet access

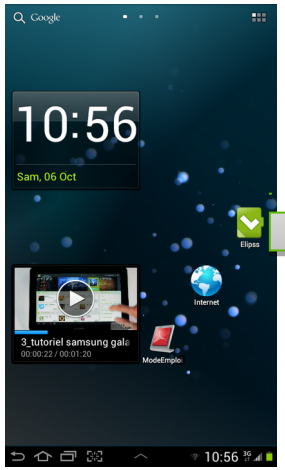
- Tablet as an incentive

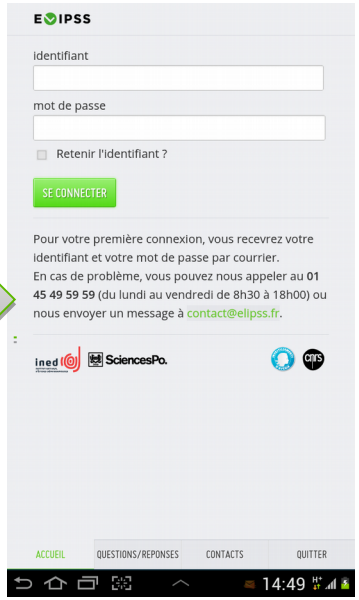
Free tablet	62 %
Trust in ELIPSS institutions	46 %
Originality of the project	37 %
Interested in research	32 %
Free internet access	13 %

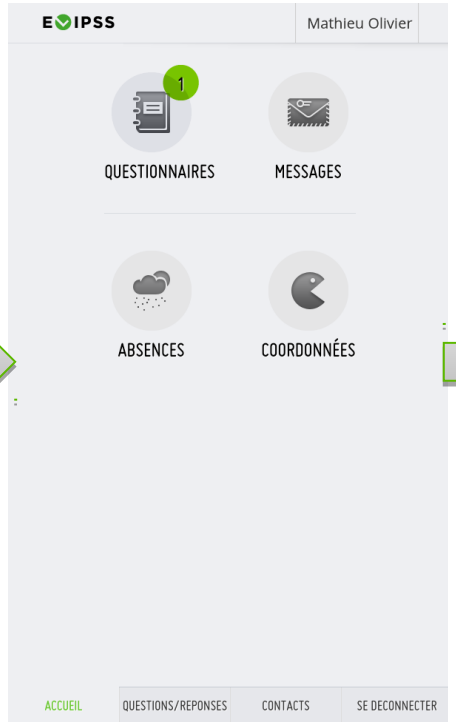
Main reasons for participating in the ELIPSS panel

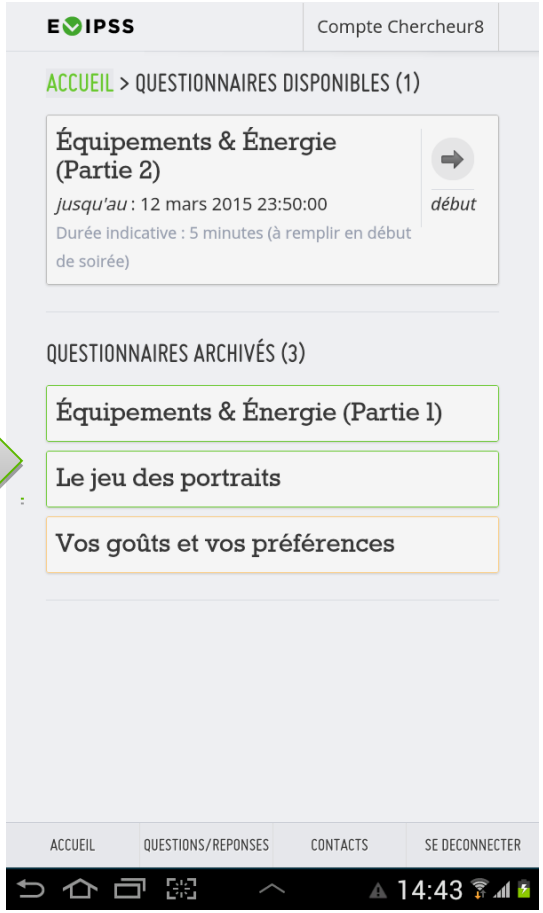
- Internet pratice change
  - 90 % of non-Internet users in 2013 are regular users in 2014

Access to Surveys









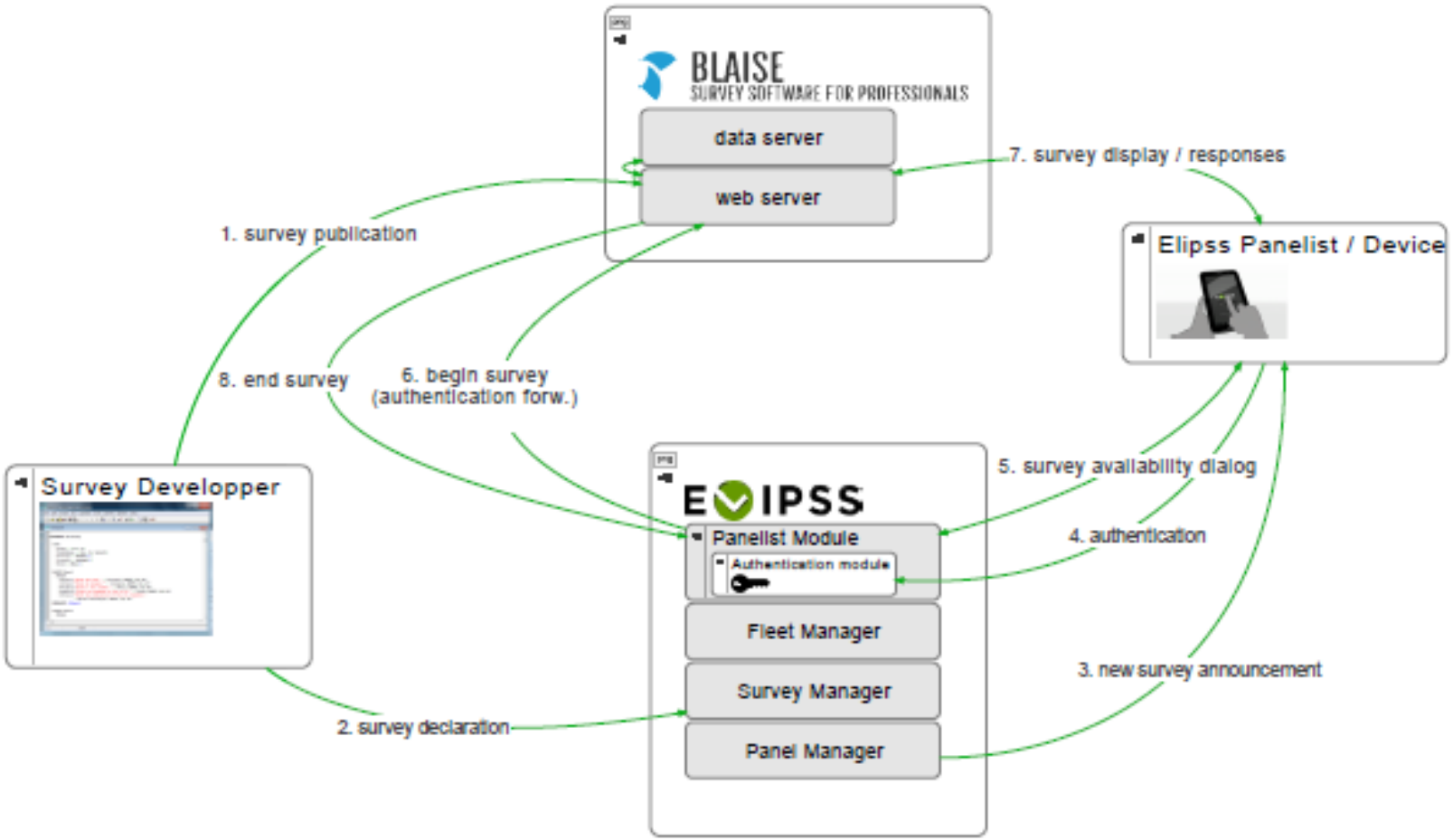
Elipss app

Log in

Home page

Available surveys

Technical Workflow

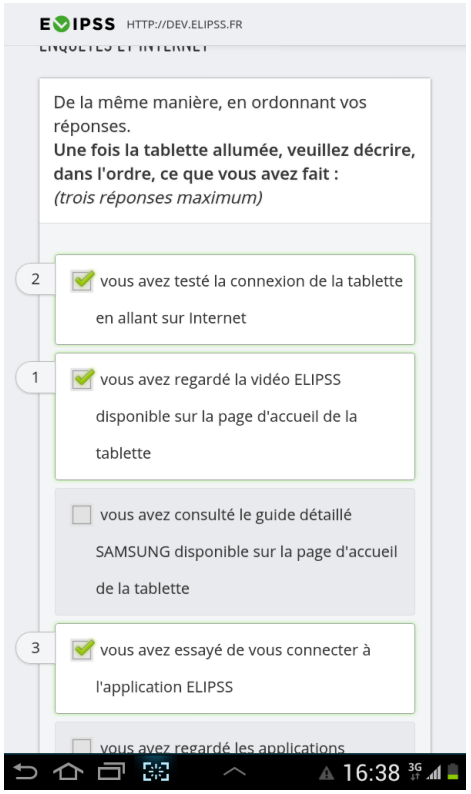


Survey design (1)

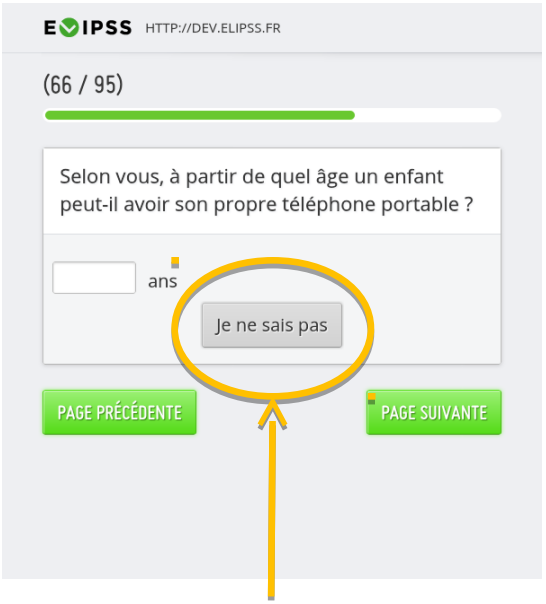
Polite probe



Enumerated type



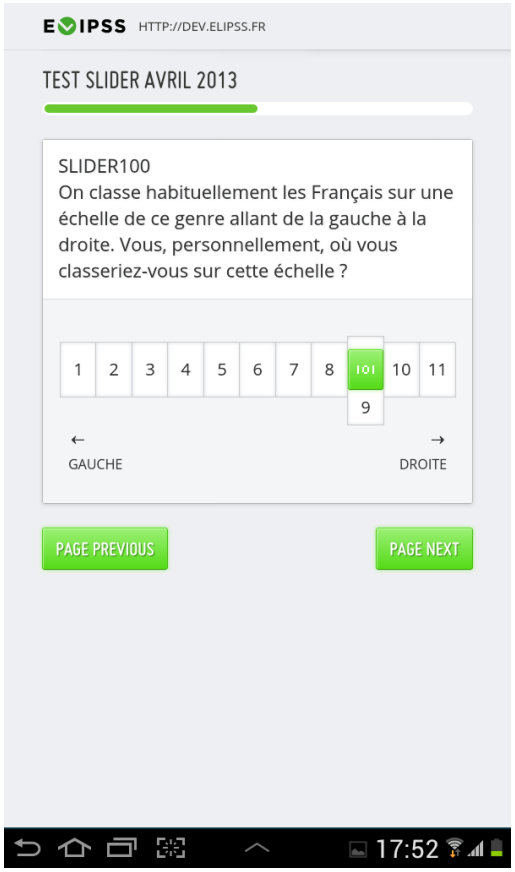
Set of enumerated-type



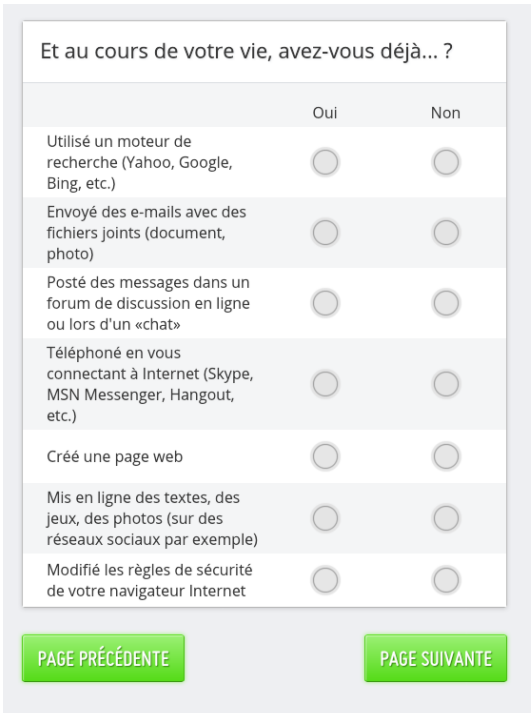
Non-response option : DK button

Integer type

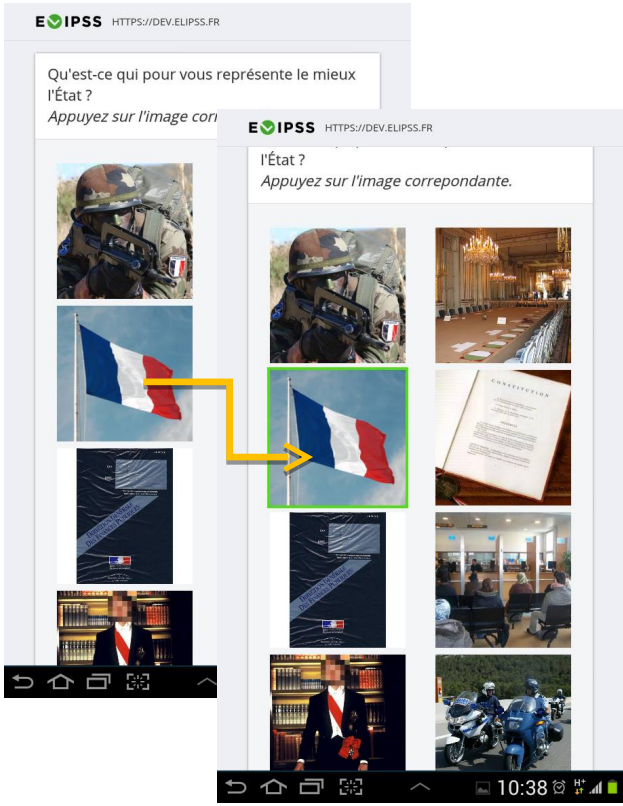
Survey design (2)



Slider



Group table



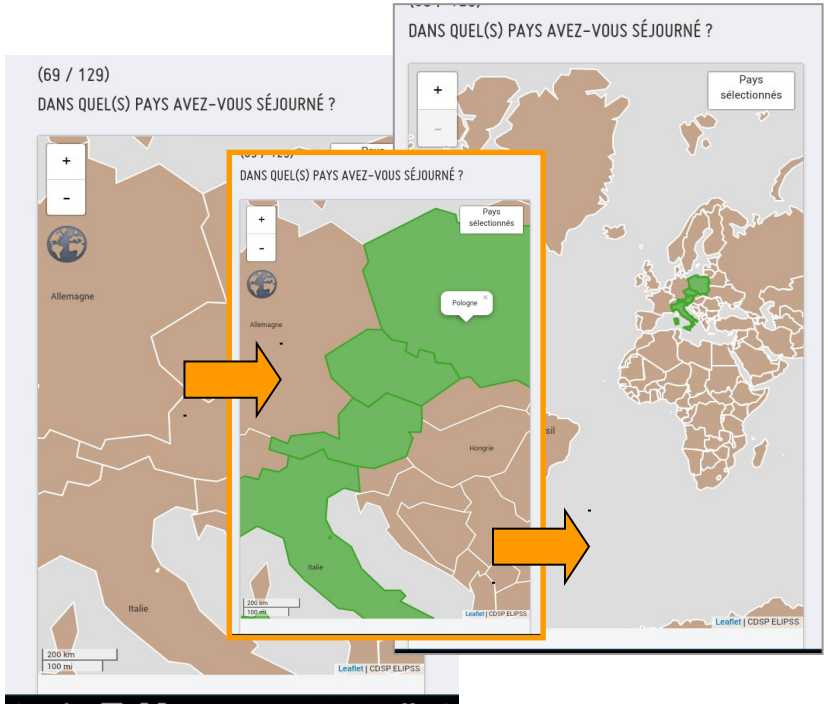
Enumeration with images

Specific features

- Technical opportunities linked to the Internet and to the tablet



Drag & drop



Clickable map

Specific feature: Photo example

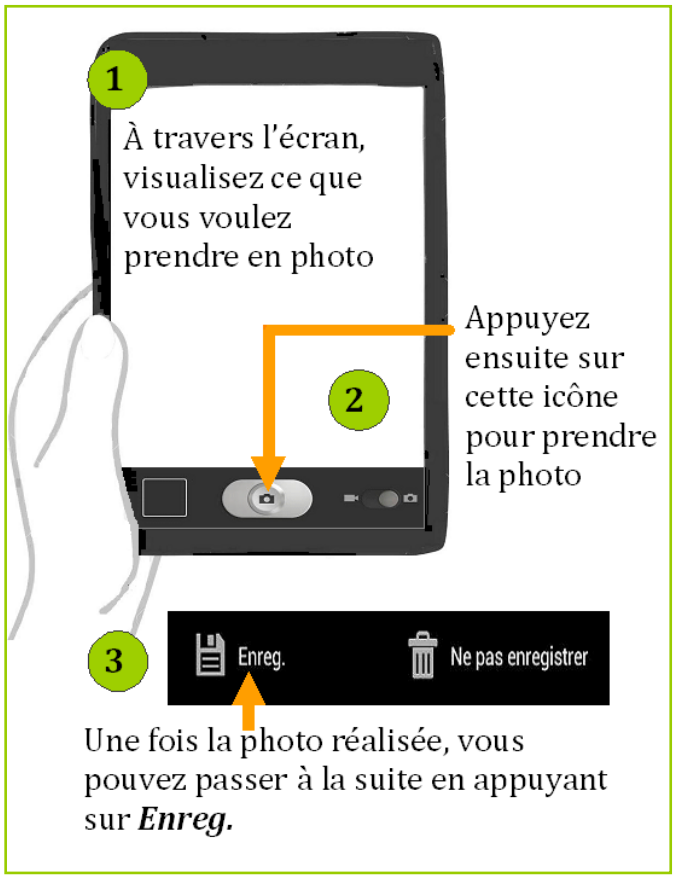


Photo tutorial

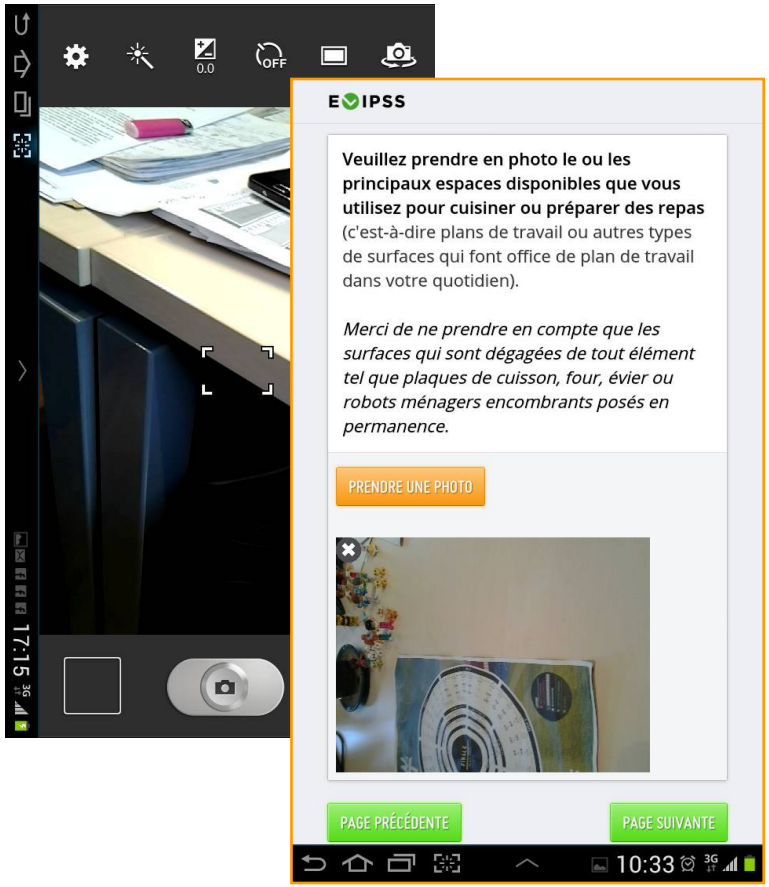


Photo feature

Photo example (1)

Javascript layer:

- Vanilla / JQuery → displaying, server communication
- Cordova → mobile functionalities

Server:

- Django 1.7.4 (Python)

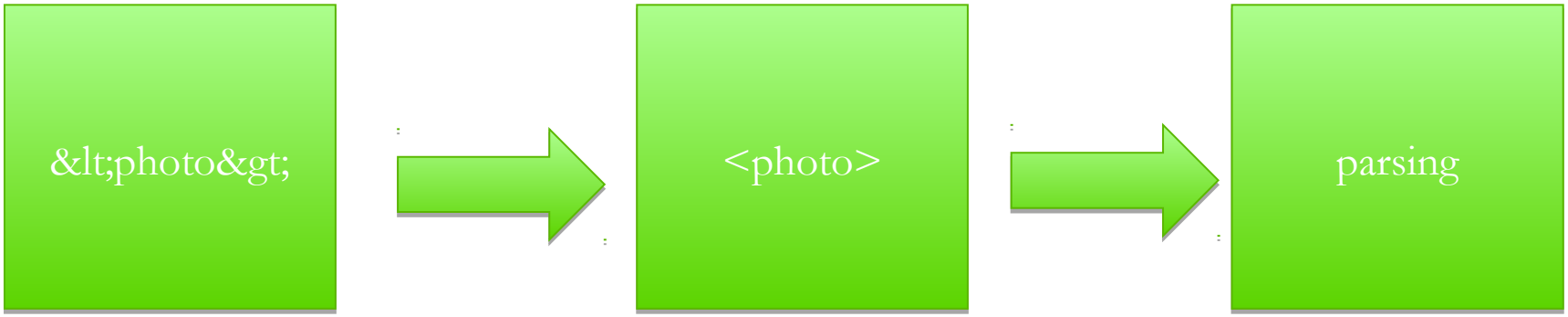
Photo example (2)

- Declare the libraries in the XSL file
- Add special tag xml in the Blaise code

```
d7 (invisible) "<photo><folder>SHAMA_201607</folder><variable>d7</variable><nb>5</nb><nb_max>6</nb_max><username>^uid</username></photo>
               @BVeuillez prendre en photo le ou les principaux espaces disponibles que vous utilisez pour cuisiner ou préparer des repas@
               @/@/@IMerci de ne prendre en compte que les surfaces qui sont dégagées de tout élément tel que plaques de cuisson, four, é
               "Espaces préparation repas" : STRING, EMPTY
```

Own XML tag in Blaise code

Photo example (3)



- &lt ; → html character less than “<”
- &gt ; → html character greater than “>”

Photo example (4)

- Parsing HTML Blaise's question
- In JQuery

```
ds.b.xml_question(

        "photo",
            [
                {'name':"variable", 'expression': "photo > variable"},
                {'name':"folder",   'expression': "photo > folder"  },
                {'name':"username", 'expression': "photo > username" },
                {'name':"limit",    'expression': 'photo > limit' },
                {'name':"nb",   'expression': 'photo > nb' },
                {'name':"nb_max",   'expression': 'photo > nb_max' },
            ],
        ds.b.utils.takephoto
    );
```

callback

Parsing XML tag

Photo example (5)

```
ds.b.xml_question = function(keyword,expressions_name,callback)
{

    var value_name = {};

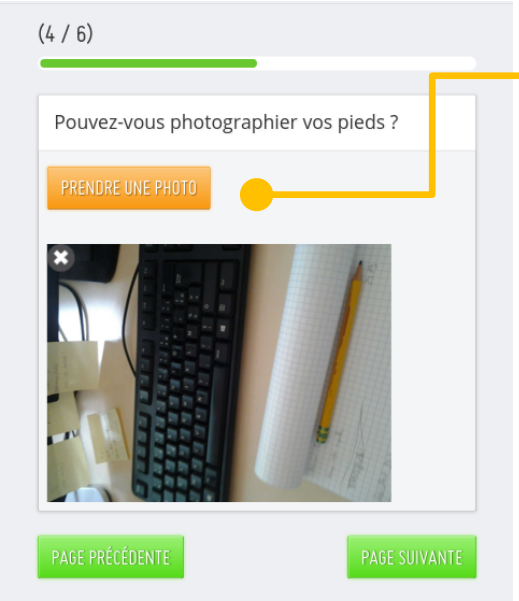
    for(var i=0;i<$("h1").length;i++) // we loop on each questions titles on
                                      // the page
    {
        var el = $("h1")[i];
        var txt = el.innerHTML;
        while(txt.indexOf('&lt;'+keyword+'&gt;')!=-1) // We check if there is
                                                      // xml string which the
                                                      // correct keyword (ex :
                                                      // <slider>)
        {
            txt = el.innerHTML;
            // We get the index of the beginning and the end of the xml str
            var ind_beg = txt.indexOf("&lt;"+keyword);
            var ind_end = txt.indexOf("&lt;/"+keyword+"&gt;")+("&lt;/"+keyword+"&gt;").length;
            var xml=txt.substring(ind_beg,ind_end);// Then we extract the xml
                                                   // from the title txt
            xml = xml.replace(/&gt;/g,'>');
            xml = xml.replace(/&lt;/g,'<');
            var xmlDoc = $.parseXML( xml );// Now we can parse the xml
            var xml = $( xmlDoc );
            // We loop in the xpath expressions_names and store the result in
            // the value_name array
            for(var j in expressions_name){
                value_name[expressions_name[j]['name']] = xml.find(expressions_name[j]['expression']).text();
                if(value_name[expressions_name[j]['name']] ==undefined)
                    return;
            }

            // We delete the xml in the title
            txt = txt.substring(0,ind_beg)+txt.substring(ind_end,txt.length);
            el.innerHTML=txt;
            callback(value_name,el,ind_beg);
        }
    }
}
```

XML parser function

Photo example (6)

- Once the XML is parsed
- Call the callback linked to this tag




```
navigator.camera.getPicture(onPhotoDataSuccess, onFail, { quality: 50,
    destinationType: Camera.DestinationType.DATA_URL });
```

```
function onPhotoDataSuccess(imageData) {
    // Uncomment to view the base64-encoded image data

    // Get image handle
    //
    var photo = document.getElementById(photoFileName+'0');
    var i=0;

    while(photo != undefined){
        i += 1;
        photo = document.getElementById(photoFileName+i);
    }

    if(typeof(Storage) !== "undefined"){
        var answer = document.getElementsByClassName('answer')[0];
        localStorage.setItem(photoFileName+i, imageData);
        var smallImage = document.createElement("img");
        smallImage.id = photoFileName+i;
        smallImage.className = "stored-photo";
        smallImage.style.width = "400px";
        smallImage.style.heigth = "600px";
        smallImage.style.display = 'block';
        smallImage.src = "" + localStorage.getItem(photoFileName+i);
```

Link between function and interface

Photo example (6)

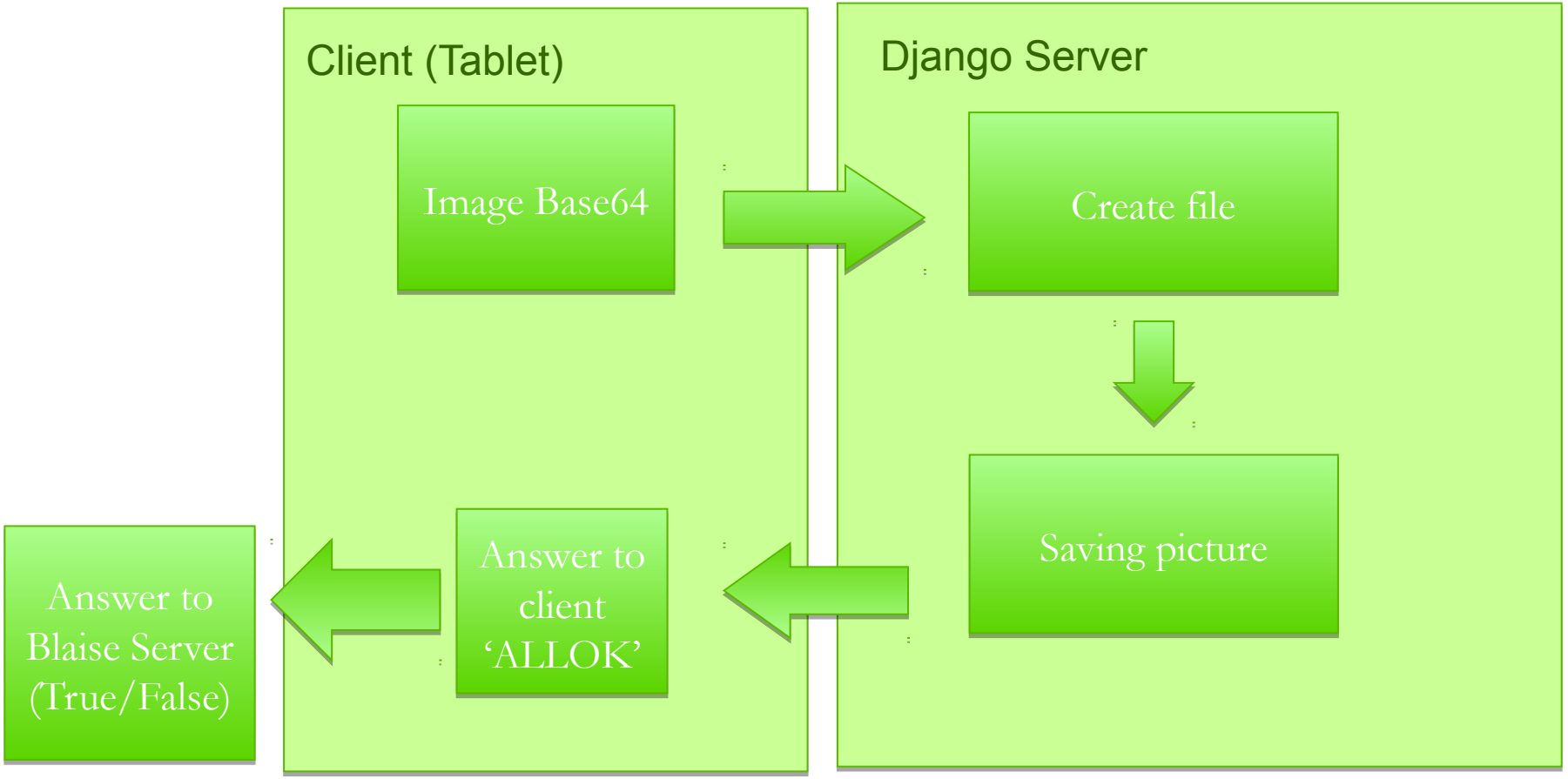


Photo example (7)

- Sending photos on the web server

```
photos_to_save = request.POST.getlist('base64_photos[]')
photos_names = request.POST.getlist('photos_names[]')
filepathparts = photos_names[0].split("-")
filepath = LOGGING_APPLICATION_PATH+"/photo/"+filepathparts[0]+"/"+filepathparts[1]+"/"+filepathparts[2]+"/"
logger.info(filepath)
logger.info(filepathparts)
#filepath = LOGGING_APPLICATION_PATH+"/photo/"+current_user.username+"/"
_assure_path_exists(filepath)
count = 0
logger.info(photos_to_save)
for photo in photos_to_save:

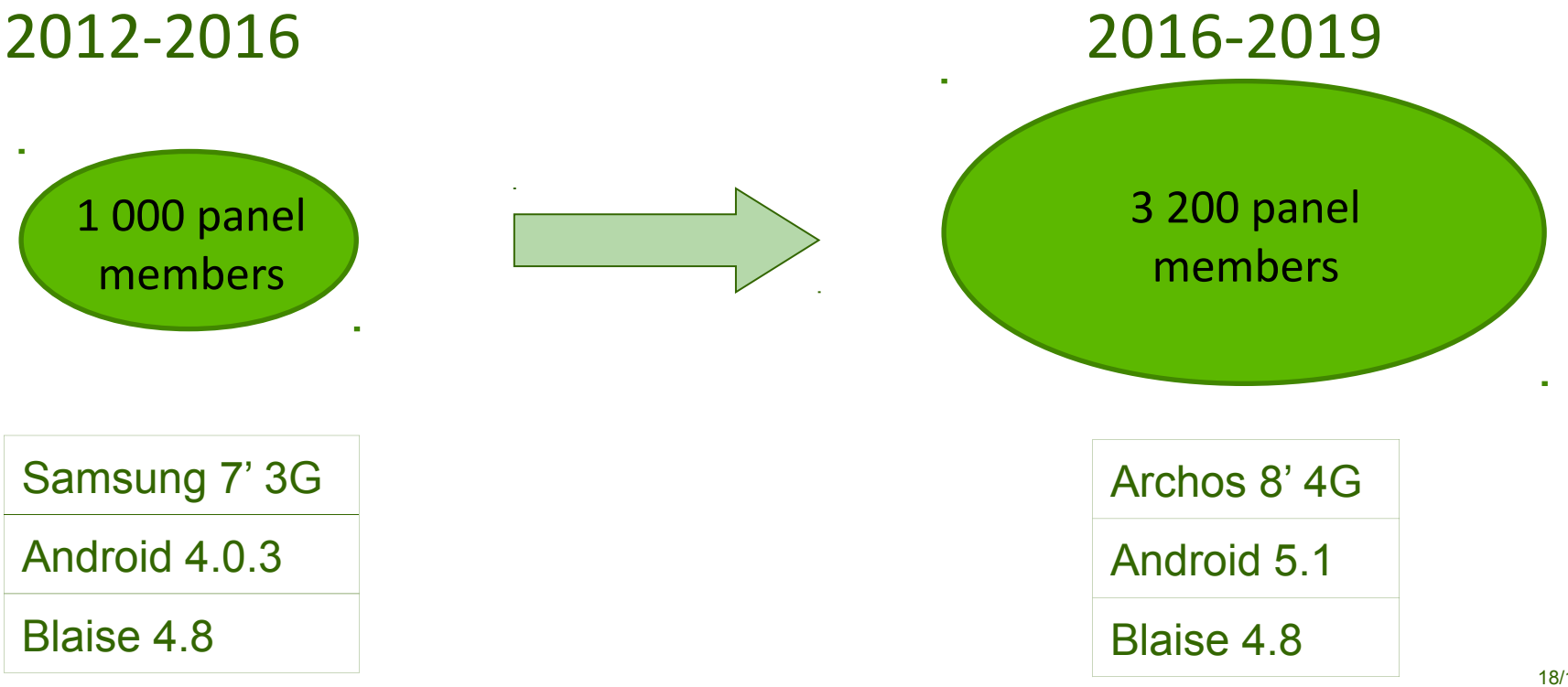
    if(count <= 5):
        imgdata = base64.b64decode(photo)
        filename = "photo-"+str(count)+".jpeg"
        filephotopath = filepath+filename

        with open(filephotopath, 'wb') as f:
            f.write(imgdata)
        count += 1
return render_to_json( response )
```

Extract from the server code

Next step

- 2 400 new panel members
- Available to the whole scientific community



Thank you for your attention!

alexandre.chevallier@sciencespo.fr
mathieu.olivier@sciencespo.fr

More informations about the project:
http://quanti.dime-shs.sciences-po.fr/