ECSP – Embedded Client Side Paradata*

Stephan Schlosser (University of Göttingen)
Jan Karem Höhne (University of Mannheim; RECSM-Universitat Pompeu Fabra)

ECSP introduction

Embedded Client Side Paradata (ECSP) is a tool that is licensed under the Creative Commons Attribution 4.0 International License (see http://creativecommons.org/licenses/by/4.0/).[1] It is based on different program languages, such as JavaScript and HTML. In general, ECSP can be implemented in web-based survey software solutions that provide access to the source code.[2] It enables researchers to passively collect different kinds of client-side paradata, such as response times and scrolling events, and data from built-in sensors, such as Global Positioning System (GPS) and acceleration data. This is irrespective of the Internet browser and operating system used and allows researchers to investigate respondents' completion behavior with respect to web surveys. Paradata and sensor data are collected at the page-level and are stored together with the actual survey data (i.e., respondents' answers) in the same dataset.

This contribution represents an updated version of the ECSP tool published by Schlosser (2016) and Schlosser and Höhne (2018). It is an expansion of the 2016 and 2018 versions that introduces five new data types. We do not discuss the contents of the previous versions from 2016 and 2018. For more detailed information on the functionality, implementation (for non-optimized survey designs), and usage of ECSP, we refer interested readers to Schlosser (2016) and Schlosser and Höhne (2018). This also applies to the descriptions of program codes published in the previous versions, including the following paradata types: 1) browser window size, 2) connection type, 3) device orientation and orientation change, 4) keystrokes (PCs and laptops only), 5) mouse clicks and finger taps, 6) mouse movements, 7) response and transmission times, 8) screen resolution and pixel ratio, 9) SurveyFocus (basic and mobile), 10) vertical and horizontal scrolling, 11) user agent strings, and 12) zooming.

*This manuscript contains the ECSP codes from 2016 and 2018 and the new codes from 2020 (see Appendix A and B). For the sake of convenience, we recommend that ECSP users only cite the new version (Schlosser & Höhne, 2020) when using the ECSP codes for their research.

Corresponding author
Stephan Schlosser, Faculty of Social Sciences, Center of Methods in Social Sciences, University of Göttingen, Goßlerstraße 19 (Room 1.103), 37073 Göttingen, Germany.
Email: sschlos1@uni-goettingen.de

In this contribution, we introduce and outline the program codes of the following five new data types: 1) acceleration data (SurveyMotion with and without gravity; see Höhne, Revilla, & Schlosser, 2020; Höhne & Schlosser, 2019), 2) compass data, 3) Global Positioning System (GPS) data, 4) gyroscope data, and 5) swiping.

The codes can be customized to suit individual needs. In other words, only data types that are deemed necessary can be collected. ECSP users can drop the codes that they do not need without affecting the collection of the remaining data types. However, the collection of response and start times is always necessary for collecting time stamps for the other data types.

ECSP extension

Acceleration data (SurveyMotion with and without gravity)

ECSP can register the acceleration of (mobile) devices, including time stamps (in milliseconds). Acceleration is defined as the rate of change of velocity of an object over time; the lower/higher the rate of change of velocity of an object in a specific time period is, the lower/higher its acceleration (Höhne & Schlosser, 2019, p. 381). Acceleration is measured in "meter per second squared (m/s^2)" and can occur on three different axes (Höhne et al., 2020, p. 44): the x-axis (i.e., left and right), the y-axis (i.e., up and down), and the z-axis (i.e., back and forth).

SurveyMotion (SMotion) measures the total acceleration of mobile devices with and without gravity (see Höhne et al., 2020; Höhne & Schlosser, 2019) to explore completion behavior in mobile web surveys. Total acceleration is defined as follows:

$$\text{Total acceleration} = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

Equation 1. Calculating total acceleration

Note. Accelerations (a) along the x-, y-, and z-axis are defined as a_x, a_y, and a_z, respectively.

For a more detailed description of SMotion with and without gravity, we refer interested readers to Höhne et al. (2020) and Höhne and Schlosser (2019).

Compass data

ECSP can register the (geographical) orientation of (mobile) devices relative to their starting position when a web survey page is completely loaded, including time stamps (in milliseconds). The values range from 0 to 360 degrees; 0 represents the starting position. Orientation changes are measured on the z-axis of the device. For instance, if a respondent turns 90 degrees to the right after the web survey page is loaded ECSP records the value 270 degrees. Thus, the angles

increase counterclockwise. It is important to note that the values registered by ECSP do not inform about the cardinal directions (i.e., north, east, south, and west).

Global Positioning System (GPS) data

ECSP can register Global Positioning System (GPS) data informing about the geographic position and position changes of (mobile) devices. ECSP collects the following five information: Latitude (in degree; e.g., 51.5608715), longitude (in degree; e.g., 10.0083852), accuracy (in meter), altitude (in meter above sea level), and time stamps (in milliseconds). The actual geographic position is determined by latitude and longitude. If there is no unobstructed line of sight to enough GPS satellites (e.g., because of mountains or high buildings) the signals might be too weak for (accurately) determining the geographic position. In order to register GPS data respondents must give their explicit consent via the Internet browser (e.g., via a pop-up window that asks respondents for their allowance to gather their geo location).

Gyroscope data

ECSP can register the rotation of (mobile) devices, including time stamps (in milliseconds). Rotation can occur on the x-axis (i.e., beta rotation), the y-axis (i.e., gamma rotation), and the z-axis (i.e., alpha rotation). The values of beta rotation range from –180 to 180, the values of gamma rotation range from –90 to 90, and the values of alpha rotation range from 0 to 360 (see https://developer.mozilla.org/en-US/docs/Web/API/DeviceOrientationEvent).

Swiping

ECSP can register swiping events (e.g., swiping right) for operating (mobile) devices. ECSP collects the following four information: Swiping (dummy variable; 0 = start and 1 = end), swiping x (coordinate on the x-axis in pixel), swiping y (coordinate on the y-axis in pixel), and time stamps (in milliseconds). During the swiping process no other operating activities (e.g., clicking) are possible or recordable. In practice, swiping can be used for objectively defining scrolling events without relying on arbitrary time thresholds, such as 100 ms between each scrolling event (Höhne & Schlosser, 2019).

ECSP limitations

The current data transfer occurs synchronically between the clients' Internet browsers and the web survey host (i.e., the server). However, an asynchronous communication via AJAX (Asynchronous JavaScript and XML) would be highly desirable because this would decrease

transmission times between clients and the host. This would enable researchers to collect information about respondents' completion behavior if they do not submit a survey page (e.g., by clicking the "Next" button) because they break-off from the web survey. Furthermore, if respondents go back and forth in the survey (e.g., using the Internet browser buttons) paradata and sensor data are overwritten.

A further limitation is that ECSP cannot collect any data if respondents have deactivated JavaScript. However, the proportion of respondents that have deactivated JavaScript is typically small. Schlosser and Höhne (2017) and Höhne, Schlosser, and Krebs (2017) have shown that this situation applies to less than 1% of all respondents, irrespective of the device type used.

ECSP and ethical considerations

The use of program languages, such as JavaScript, enables researchers to passively collect many data without respondents' knowledge and consent, which also applies to the collection of client-side paradata and sensor data by means of ECSP. Researchers using such data face ethical considerations. Although we encourage researchers to use paradata and sensor data to improve survey research methods, we clearly state that these data should not be used to surveil respondents or to frivolously adapt responses given by respondents (Heerwegh, 2002). We are convinced that these kinds of data should not be collected without respondents' consent, even if willingness to participate decreases (Couper & Singer, 2013). Furthermore, we highly recommend checking (specific) legal prerequisites to protect the online privacy of respondents.

ECSP disclaimer

Although the authors tested the application of ECSP in several (pretest) studies, they wish to state clearly here that the use of all program codes is completely the user's own responsibility. There is no warranty of any kind that the codes work properly, and users are encouraged to test their functionality before utilization. The authors and/or their affiliations cannot be held responsible for any malfunctions and/or damages, even if ECSP is the responsible source.

ECSP implementation for optimized survey designs[3]

We now describe a seven-step procedure to implement ECSP in the survey software solution Unipark (Questback). This implementation procedure includes the following steps: 1) implementing JavaScript code (see Appendix A), 2) generating an invisible user-defined question, 3) generating paradata and sensor data variables, 4) implementing HTML code (see Appendix B), 5) adapting HTML code, 6) connecting survey page and JavaScript, and

7) functionality test. All steps should be conducted carefully and in the given order. It is recommended that the ECSP tool is implemented after the questionnaire has been programmed.

Steps 1 to 5 must be conducted only for the first questionnaire page because the invisible user-defined question can be simply copied for all the other pages. However, step 6 must be conducted manually for each questionnaire page.

Step 1: Implementing JavaScript code

Select *Page structure of standard questionnaire pages* by accessing the sub menu *Pro editor* (*Layout → Pro editor*).

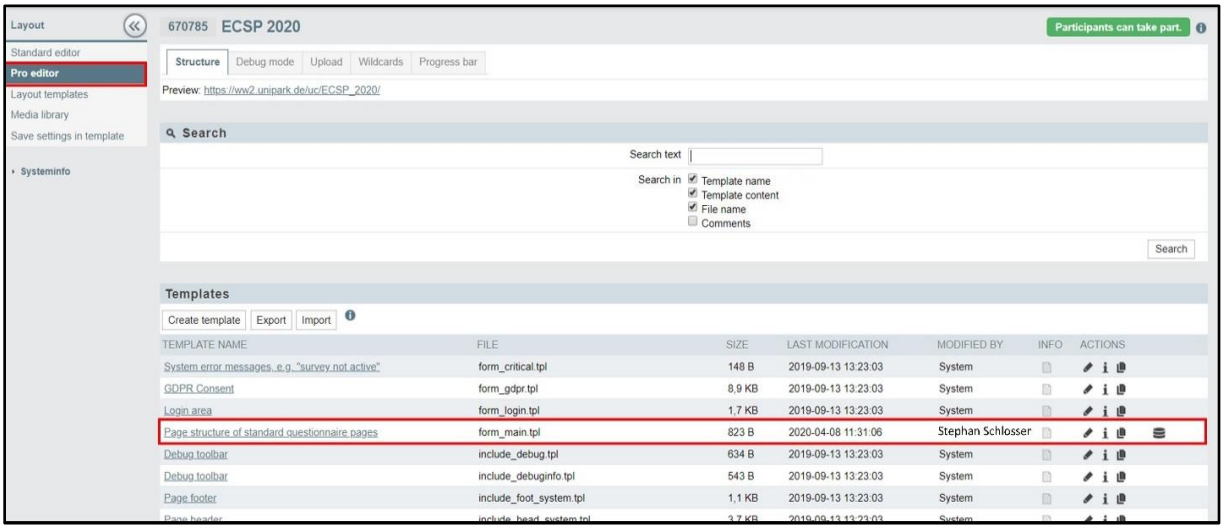

Figure 1. Pro editor and page structure of standard questionnaire pages

Implement the ECSP JavaScript code (see Appendix A) in the source code and click the *Save* button. Do not adapt the main code.

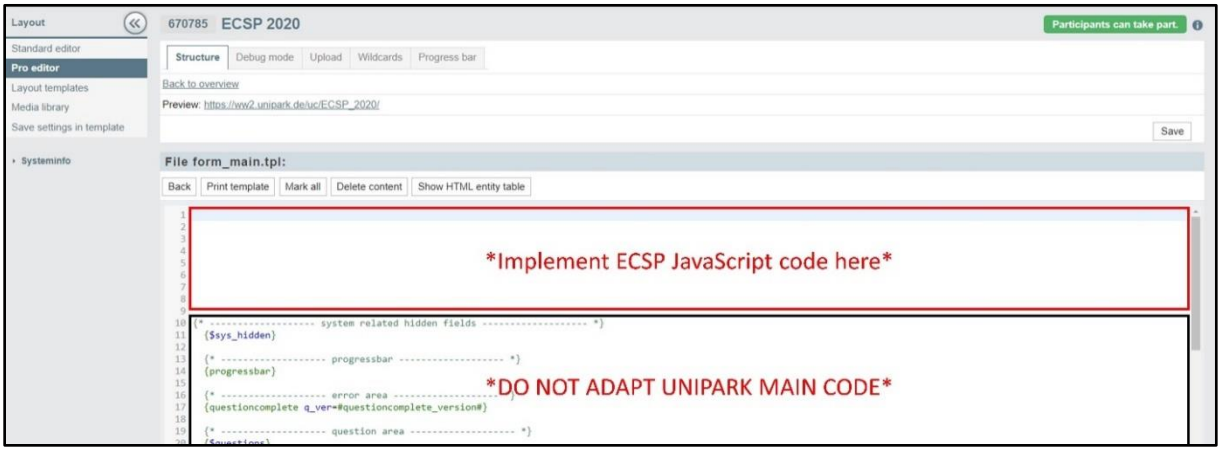

Figure 2. Access point to the JavaScript source code

Step 2: Generating an invisible user-defined question

Access the sub menu *Questionnaire editor* and generate a further question (type: *911 User-defined*) for the first questionnaire page.

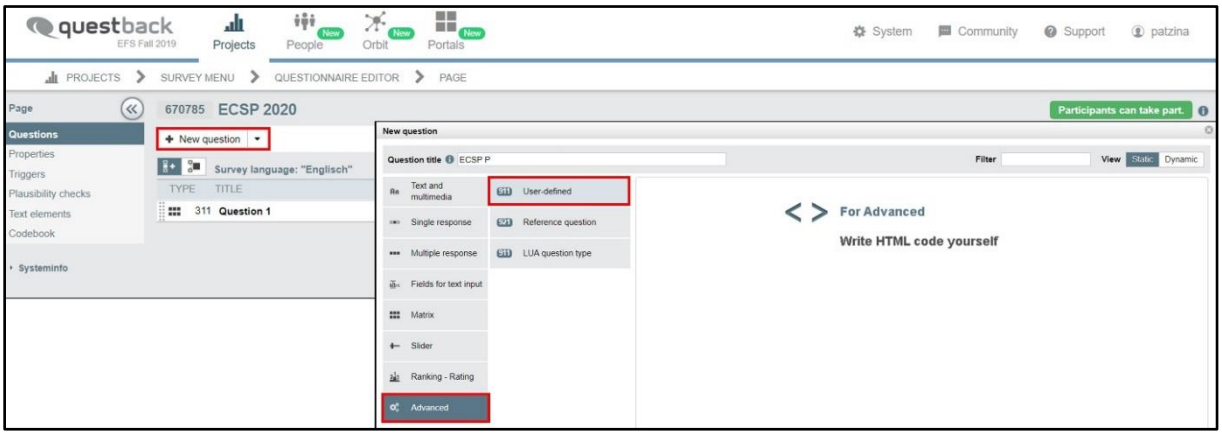

Figure 3. Generating an invisible user-defined question

Step 3: Generating paradata and sensor data variables

For each paradata and sensor data type, several variables must be generated. For example, to collect SurveyMotion (SMotion) without gravity, the following variables must be generated:

1) SURVEYMOTION (SMOTION) WITHOUT GRAVITY
2) SURVEYMOTION (SMOTION) WITHOUT GRAVITY TIME

The variable type *Text (max. 65535 characters)* must be used for all paradata and sensor data types collected by ECSP.

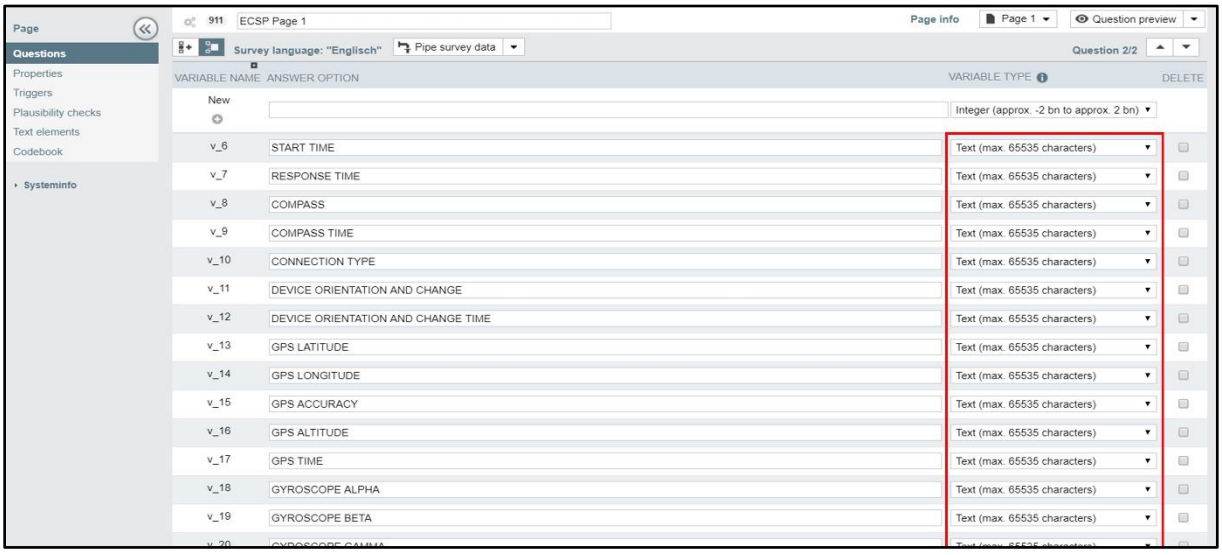

Figure 4. Paradata and sensor data variables

Step 4: Implementing HTML code

Go to the *Edit HTML* section at the bottom of the *911 User-defined* question page. Implement the ECSP HTML code (see Appendix B).

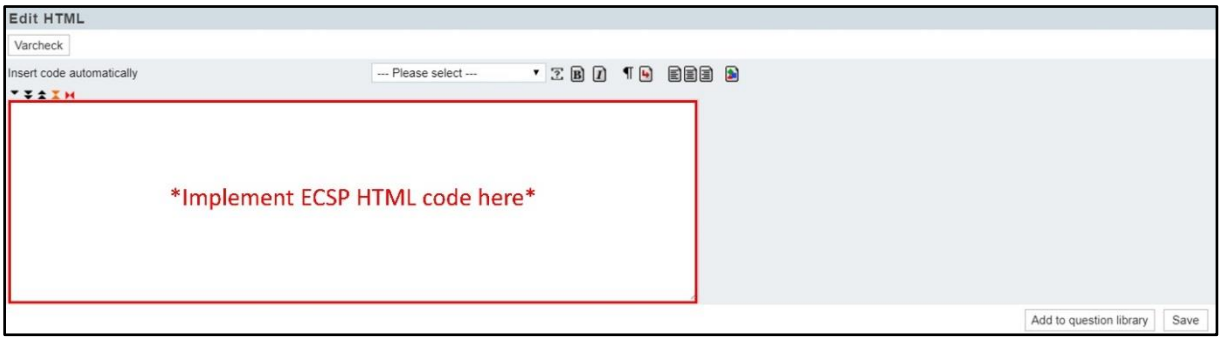

Figure 5. HTML section

Step 5: Adapting HTML code

Adapt the correct variable names given by Unipark (Questback) in the ECSP HTML code. This adaption must be accomplished for all paradata and sensor data variables. To check the correctness of the variable names, click the *Varcheck* button. Click the *Save* button at the end.

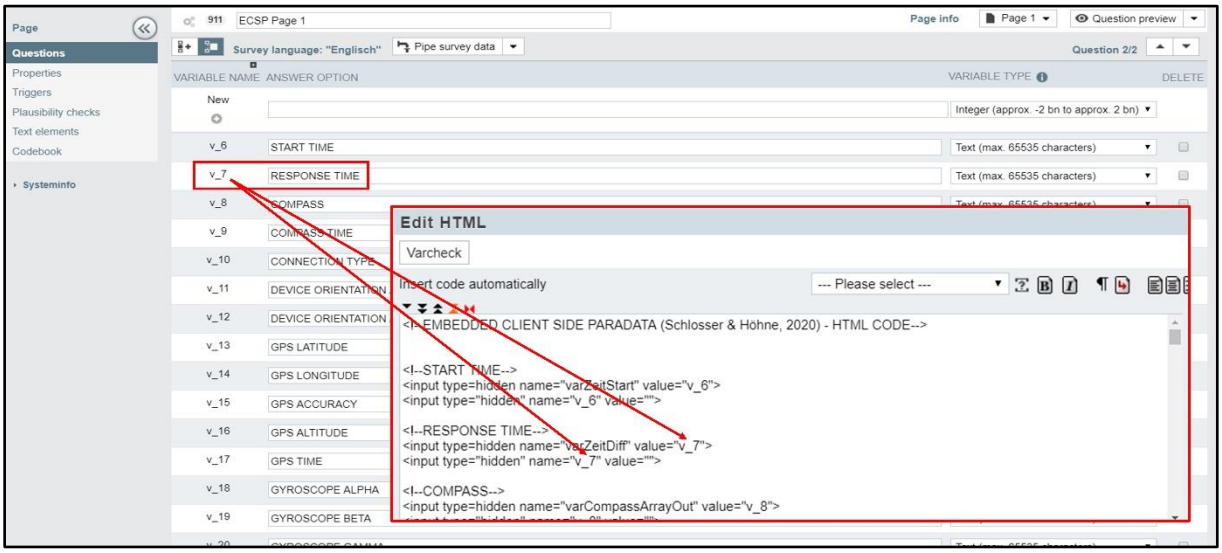

Figure 6. Editing ECSP HTML code

Step 6: Connecting survey page and JavaScript

Access the sub menu *Properties* on the *911 User-defined* question page. Go to the *Additional code* section and enter *send();* in the input field and click the *Save* button.

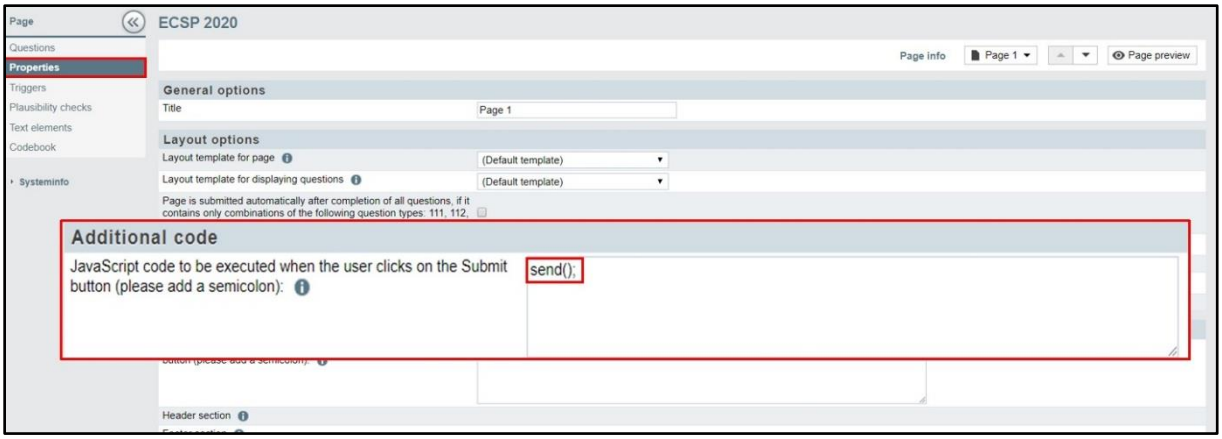

Figure 7. Additional code section

This step must be repeated manually for each *911 User-defined* question page.

Step 7: Functionality test

Conducting functionality tests is highly recommended to make sure that the paradata and sensor data collection by ECSP is working properly. For example, complete the questionnaire with your smartphone and trigger different paradata and sensor data events (e.g., changing the device orientation) and control the results.

Endnotes

[1] Embedded Client Side Paradata (2020) by Stephan Schlosser (University of Göttingen) and Jan Karem Höhne (University of Mannheim; RECSM-Universitat Pompeu Fabra) is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, please visit http://creativecommons.org/licenses/by/4.0/.

[2] ECSP was conceptualized for implementation in Unipark (Questback), which is the main survey software solution that the authors use. Therefore, the ECSP implementation procedure is exclusively for Unipark (Questback). Users should keep in mind that they might need to make some adaptions if they wish to use the ECSP program codes for other survey software solutions.

[3] See Schlosser (2016) for ECSP implementation in non-optimized survey designs. All descriptions are based on Unipark (Questback) version 10.9 (Fall 2019).

References

Couper, M. P., & Singer E. (2013). Informed consent for web paradata use. *Survey Research Methods, 7*, 57–67.

Heerwegh, D. (2002). Describing response behavior in web surveys using client side paradata. *Paper presented at the International Workshop on Web Surveys*, Mannheim: Germany.

Höhne, J. K., Revilla, M., & Schlosser, S. (2020). Motion instructions in surveys: Compliance, acceleration, and response quality. *International Journal of Market Research, 62*, 43–57.

Höhne, J. K., & Schlosser, S. (2019). SurveyMotion: What can we learn from sensor data about respondents' completion and response behavior in mobile web surveys? *International Journal of Social Research Methodology, 22*, 379–391.

Höhne, J. K., Schlosser, S., & Krebs, D. (2017). Investigating cognitive effort and response quality of question formats in web surveys using paradata. *Field Methods, 29*, 365–382.

Schlosser, S. (2016). Embedded Client Side Paradata (ECSP). *Zenodo*. DOI: 10.5281/zenodo.55169.

Schlosser, S., & Höhne, J. K. (2018). Embedded Client Side Paradata (ECSP). *Zenodo*. DOI: 10.5281/zenodo.1218941

Schlosser, S. & Höhne, J. K. (2017). Does the continuity of web-survey processing matter? *Paper presented at the Conference of the European Survey Research Association*, Portugal: Lisbon.

Biographical note

Stephan Schlosser (sschlos1@uni-goettingen.de) is a doctoral candidate and research associate at the Center of Methods in Social Sciences at the University of Göttingen, Germany. His research combines survey methodology, computer science, and data science.

Jan Karem Höhne (hoehne@uni-mannheim.de) is a postdoctoral researcher at the Collaborative Research Center 884 "Political Economy of Reforms" at the University of Mannheim, Germany, and research fellow at the "Research and Expertise Centre for Survey Methodology" at the Universitat Pompeu Fabra in Barcelona, Spain. His research combines survey methodology, computer science, and data science.

Acknowledgment

The authors would like to thank Tobias Baier (Darmstadt University of Technology), Mick P. Couper (University of Michigan), Anne Elevelt (Utrecht University), Christoph Kern (University of Mannheim), Tanja Kunz (GESIS – Leibniz Institute for the Social Sciences), Daniel Qureshi (University of Frankfurt), and Melanie Revilla (RECSM-Universitat Pompeu Fabra) for their cooperation, inspiration, and support.

Appendix A

In the following, we share the ECSP JavaScript codes from 2016, 2018, and 2020. Before using ECSP, we highly recommend reading the chapters “ECSP and ethical considerations” and “ECSP disclaimer” above. We listed all codes in an alphabetical order, except for response and start times because they need to be collected first.

```
{*EMBEDDED CLIENT SIDE PARADATA (Schlosser & Höhne, 2020) - JAVASCRIPT
CODE*}


<html>
<head>{$sys_head}

{* JQUERY *}
<script type='text/javascript'
src="//ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">'></script
>

</head>

<title>{$msg_title}</title>
<script type='text/javascript'>

{literal}

jQuery.noConflict();


//RESPONSE AND START TIME (1)

var now=(function(){
     var performance=window.performance || {};
     performance.now=(function(){
     return performance.now ||
     performance.webkitNow ||
     performance.msNow ||
     performance.oNow ||
     performance.mozNow ||
     function(){return new Date().getTime();};
     })();
     return performance.now();
});
var start=now();
x=new Date();


//COMPASS

var CompassArray=new Array();
var CompassTArray=new Array();

window.addEventListener('deviceorientation', handleDeviceOrientation,
true);

if(window.DeviceOrientationEvent){
     window.addEventListener("deviceorientation", function(event){
         var alpha2;
```

```
            var timeCompass;
            if(event.webkitCompassHeading){
                alpha2=event.webkitCompassHeading;
            }
            else{
                alpha2=event.alpha;
                if(!window.chrome){
                    alpha2=alpha2-270;
                }
            }
            var timeCompass=Math.round(now()-start);
            handleOrientationEvent(alpha2, timeCompass);
        }, true);
}

var handleOrientationEvent=function(alpha2, timeCompass){
        CompassArray[CompassArray.length]=alpha2.toFixed(0);
        CompassTArray[CompassTArray.length]=timeCompass;
};


//CONNECTION TYPE

function connection_type(){
        "use strict";
        var strOnError, strConnection, strOut;
        strOnError="N/A"; strConnection=null;
        strOut=null;
        try{
            strConnection=navigator.connection.type;
            strOut=strConnection;
        }
        catch(err){
            return strOnError;
        }
        return strOut;
}
var ConnectionType;

(function($){
        $(document).ready(function(){
            ConnectionType=connection_type();
        });
})(jQuery);


//DEVICE ORIENTATION AND ORIENTATION CHANGE

var ScreenTArray=new Array();
var ScreenOrientArray=new Array();

var mo={
        _browser: null,
        _os: null,
        _ua: navigator.userAgent,
        normalise: false,
        orientation: false,
        motion: false,
        init: function(){
            var orientation=false;
            var motion=false
            if(window.DeviceOrientationEvent){
```

```
            orientation=true;
        }
        if(window.DeviceMotionEvent){
            motion=true;
        }
        if(orientation && motion){
            if(this._ua.match(/Firefox/i) && this._ua.match(/Android/i)){
                this._os="Android";
                this._browser="Firefox";
            } else if(this._ua.match(/Android/i)){
                this._os="Android";
                this._browser="Stock";
            } else if(this._ua.match(/Blackberry|RIM/i)){
                this._os="Blackberry";
                this._browser="Stock";
            } else{
                this._os="iOS";
                this._browser="webkit";
            }
        } else if(orientation && !motion){
            this._browser="Chrome";
            if(this._ua.match(/Android/i)){
               this._os="Android";
            } else{
                this._os="Desktop";
            }
        } else if(!orientation){
            this._browser="Unknown";
            this._os="Unknown";
        }
        this.orientation=orientation;
        this.motion=motion;
    },
};

(function($){
    function onresize2(){
        ScreenTArray[ScreenTArray.length]=Math.round(now()-start);
        ScreenOrientArray[ScreenOrientArray.length]=window.orientation;
    };
    $(window).resize(onresize2);
    onresize2();
})(jQuery);


//GPS

var latitudeArray=new Array();
var longitudeArray=new Array();
var accuracyArray=new Array();
var altituArray=new Array();
var GPStimeArray=new Array();

function GPSget(){
    navigator.geolocation.getCurrentPosition(
    successCallback,
    errorCallback_highAccuracy,
    {maximumAge:1000, timeout:1000, enableHighAccuracy: true}
    );
    function errorCallback_highAccuracy(position){
        if(error.code==error.TIMEOUT){
            navigator.geolocation.getCurrentPosition(
```

```
                successCallback,
                errorCallback_lowAccuracy,
                {maximumAge:1000, timeout:1000, enableHighAccuracy:
false});
         return;
         }
     }
     function successCallback(position){
         latitudeArray[latitudeArray.length]=position.coords.latitude;
         longitudeArray[longitudeArray.length]=position.coords.longitude;


accuracyArray[accuracyArray.length]=Math.round(position.coords.accuracy);

     altituArray[altituArray.length]=Math.round(position.coords.altitude);
     GPStimeArray[GPStimeArray.length]=Math.round(now()-start);
     }
     setTimeout(GPSget, 1000);
}
GPSget();


//GYROSCOPE

var AlphaArray=new Array();
var BetaArray=new Array();
var GammaArray=new Array();
var GyroTimeArray=new Array();

window.addEventListener('deviceorientation', handleDeviceOrientation,
true);

function handleDeviceOrientation(event){
     var alpha=event.alpha; //z axis rotation [0,360)
     var beta=event.beta; //x axis rotation [-180, 180]
     var gamma=event.gamma; //y axis rotation [-90, 90]
     AlphaArray[AlphaArray.length]=alpha.toFixed(4);
     BetaArray[BetaArray.length]=beta.toFixed(4);
     GammaArray[GammaArray.length]=gamma.toFixed(4);
     GyroTimeArray[GyroTimeArray.length]=Math.round(now()-start);
}


//KEY STROKES

var KeyTArray=new Array();
var KeyCodeArray=new Array();
var event='';

(function($){
document.onkeypress=function(k){
     var k=window.event || k;
     KeyCodeArray[KeyCodeArray.length]=(k.which || k.keyCode);
     KeyTArray[KeyTArray.length]=Math.round(now()-start);
     };
})(jQuery);


//MOUSE CLICK AND FINGER TAB

var ClickXArray=new Array();
var ClickYArray=new Array();
```

```
var ClickTArray=new Array();
var ClickXtext='';
var ClickYtext='';
var ClickTtext='';

(function($){
     $(function(){
     $(document.body).mousedown(function(f){
         ClickXArray[ClickXArray.length]=Math.round(f.pageX);
         ClickYArray[ClickYArray.length]=Math.round(f.pageY);
         ClickTArray[ClickTArray.length]=Math.round(now()-start);
     });
});
})(jQuery);


//MOUSE MOVEMENT

var XArray=new Array();
var YArray=new Array();
var TArray=new Array();
var Xtext='';
var Ytext='';
var Ttext='';

(function($){
$(document).ready(function(){
     $(document.body).mousemove(function (e){
         if(e.pageX !== 0 && e.pageY !== 0){
             XArray[XArray.length]=Math.round(e.pageX);
             YArray[YArray.length]=Math.round(e.pageY);
             TArray[TArray.length]=Math.round(now()-start);
         }
     });
});
})(jQuery);


//SCREEN RESOLUTION AND PIXEL RATIO

var ScreenWArray;
var ScreenHArray;
var ScreenRatioArray;

ScreenWArray=screen.width;
ScreenHArray=screen.height;
ScreenRatioArray=window.devicePixelRatio || 1;


//SCROLLING (HORIZONTAL)

var ScrollHTArray=new Array();
var ScrollHXArray=new Array();
var scrollH='';

(function($){
$(window).scroll(function(){
     var scrollH=$(document).scrollLeft();
     ScrollHXArray[ScrollHXArray.length]=scrollH;
     ScrollHTArray[ScrollHTArray.length]=Math.round(now()-start);
});
})(jQuery);
```

```
//SCROLLING (VERTICAL)

var ScrollVTArray=new Array();
var ScrollVYArray=new Array();
var scrollV='';

(function($){
$(window).scroll(function(){
     var scrollV=$(document).scrollTop();
     ScrollVYArray[ScrollVYArray.length]=scrollV;
     ScrollVTArray[ScrollVTArray.length]=Math.round(now()-start);
});
})(jQuery);


//BASIC SURVEYFOCUS (SF)

var FocusTArray=new Array();
var FocusArray=new Array();
var state='';

var visibilityChange=(function(window){
     var inView=true;
     return function(fn){

window.onfocus=window.onblur=window.onpageshow=window.onpagehide=function(s
){
     if({focus:1, pageshow:1}[s.type]){
         if(inView) return;
         fn("1"); //visible
         inView = true;
         }
         else if(inView){
             fn("0"); //hidden
             inView=false;
             }
         };
     };
}(this));

visibilityChange(function(state){
     FocusArray[FocusArray.length]=state;
     FocusTArray[FocusTArray.length]=Math.round(now()-start);
});


//MOBILE SURVEYFOCUS (SF)

var FocusTArray2=new Array();
var FocusArray2=new Array();
var state2='';
var mobout=false;

var FocusCheckr=function(){
     var _that={},
         _lastSeen=0,
         _timerInterval=10,
         _handlers=[];
         _that.timerThreshold=10;
         _that.onFocus=function(handler, params){
```

```
        var hiddenProp=getHiddenProp();
        if(hiddenProp){
            var evtName=hiddenProp.replace(/[H|h]idden/, "") +
"visibilitychange";
            document.addEventListener(evtName, function(e){
                if(isHidden()){
                    handler(e, params);
                }
            }, false);
        }else{
            var handlerObj={"handler": handler};
            if(params!==undefined){handlerObj.params=params}
            _handlers.push(handlerObj);
            startLoop();
        }
    };
    _that.offFocus=function(handler, params){
        var hiddenProp=getHiddenProp();
        if(hiddenProp){
            var evtName=hiddenProp.replace(/[H|h]idden/, "") +
"visibilitychange";
            document.addEventListener(evtName, function(e){
                if(!isHidden()){
                    handler(e, params);
                }
            }, false);
        }
    };
    var animFrame=(function(req, can){
        var af={},
        lastTime=0,
        vendors=["webkit", "moz", "o", "ms"];
        af[req]=window.requestAnimationFrame;
        af[can]=window.cancelAnimationFrame;
        for(var x=0, l=vendors.length; x < l && !af[req]; ++x){
            af[req]=window[vendors[x]+'RequestAnimationFrame'];
            af[can]=window[vendors[x]+"CancelAnimationFrame"] ||
                window[vendors[x]+"CancelRequestAnimationFrame"];
        }
        if(!af[req]){
            af[req]=function(callback, element){
                var currTime=Date.now(),
                    timeToCall=Math.max(0, 16-(currTime-lastTime)),
                    id=window.setTimeout(function(){
                        callback(currTime+timeToCall);
                    }, timeToCall);
                lastTime=currTime+timeToCall;
                return id;
            };
        }
        if(!af[can]){
            af[can]=function(id){
                clearTimeout(id);
            };
        }
        return af;
    }("request", "cancel"));
    getHiddenProp=function(){
        var prefixes=["webkit", "moz", "o", "ms"],
            prop="";
        if("hidden" in document){return "hidden"}
        for(var i=0, l=prefixes.length; i < l; i++){
```

```
                prop=prefixes[i] + "Hidden";
                if((prop) in document){return prop};
            }
            return null;
        },
        isHidden=function(){
            return document[getHiddenProp()] || false;
        },
        startLoop=function(){
            _lastSeen=Date.now();
            window.onscroll=onScrollHandler;
            animFrame.request(rafHandler);
        },
        checkFocus=function(){
            if(Date.now()- _lastSeen > _that.timerThreshold){
                notifyHandlers();
            }
            _lastSeen=Date.now();
        },
        rafHandler=function(){
            animFrame.request(rafHandler);
            checkFocus();
        },
        notifyHandlers=function(){
            var numHandlers=_handlers.length,
                handlerObj=null;
            if(numHandlers){
                for(var i=0; i < numHandlers; i++){
                    handlerObj=_handlers[i];
                    if(handlerObj["params"]){
                        handlerObj.handler(null, handlerObj["params"]);
                    } else{
                        handlerObj.handler(null);
                    }
                }
            }
        },
        onScrollHandler=function(e){
            _lastSeen=Date.now();
        };
        return _that;
};

var focusCheck=new FocusCheckr(), count=0;
focusCheck.timerThreshold=10;

focusCheck.onFocus(function(){
        FocusTArray2[FocusTArray2.length]=Math.round(now()-start);
        FocusArray2[FocusArray2.length]="0";
        mobout=true;
});

focusCheck.offFocus(function(){
        if(mobout==true){
        FocusTArray2[FocusTArray2.length]=Math.round(now()-start);
        FocusArray2[FocusArray2.length]="1";
        mobout=false;
        }
});


//SURVEYMOTION (SMOTION) WITH AND WITHOUT GRAVITY
```

```
var SM_T_Array=new Array();
var SM_Array=new Array();
var SM_G_Array=new Array();
var SM_G_T_Array=new Array();

function SurveyMotion(e){
     this.acceleration=new Object();
     if(e.acceleration!==null){

this.acceleration.x=Math.sqrt(Math.pow(e.acceleration.x,2)+Math.pow(e.accel
eration.y,2)+Math.pow(e.acceleration.z,2)).toFixed(2);

this.acceleration.g=Math.sqrt(Math.pow(e.accelerationIncludingGravity.x,2)+
Math.pow(e.accelerationIncludingGravity.y,2)+Math.pow(e.accelerationIncludi
ngGravity.z,2)).toFixed(2);
         this.acceleration.t=Math.round(now()-start);
     } else{
         this.acceleration.x=null;
         this.acceleration.g=null;
     }
     this.interval=null;
     if(e.interval!==null){this.interval=e.interval;}
         return(this);
};

window.addEventListener("devicemotion", update_accel, false);

function update_accel(e){
     var a=SurveyMotion(e);
     SM_T_Array[SM_T_Array.length]=a.acceleration.t;
     SM_Array[SM_Array.length]=a.acceleration.x;
     SM_G_T_Array[SM_G_T_Array.length]=a.acceleration.t;
     SM_G_Array[SM_G_Array.length]=a.acceleration.g;
};


//SWIPING

var SwipeArray=new Array();
var SwipeXArray=new Array();
var SwipeYArray=new Array();
var SwipeTArray=new Array();

var T_01=null;
var T_StartX=null;
var T_StartY=null;
var T_StartT=null;
var T_EndX=null;
var T_EndY=null;
var T_EndT=null;

document.addEventListener('touchstart', SwipingStart);
document.addEventListener('touchend', SwipingEnd);

function SwipingStart(evt){
     T_01=1;
     T_StartT=Math.round(now()-start);
     T_StartX=Math.round(evt.touches[0].clientX);
     T_StartY=Math.round(evt.touches[0].clientY);
}
```

```
function SwipingEnd(evt){
     T_01=0;
     T_EndT=Math.round(now()-start);
     T_EndX=Math.round(evt.changedTouches[0].clientX);
     T_EndY=Math.round(evt.changedTouches[0].clientY);
     if(T_StartT<T_EndT&&((Math.abs(T_StartX-
T_EndX)>10)||(Math.abs(T_StartY-T_EndY)>10))){
          SwipeArray[SwipeArray.length]=T_01;
          SwipeTArray[SwipeTArray.length]=T_StartT;
          SwipeXArray[SwipeXArray.length]=T_StartX;
          SwipeYArray[SwipeYArray.length]=T_StartY;
          SwipeArray[SwipeArray.length]=T_01+1;
          SwipeTArray[SwipeTArray.length]=T_EndT;
          SwipeXArray[SwipeXArray.length]=T_EndX;
          SwipeYArray[SwipeYArray.length]=T_EndY;
     }
};


//USER AGENT STRING

var UserAgentJS;
UserAgentJS=window.navigator.userAgent;


//WINDOW SIZE AND ZOOM

var SizeXArray=new Array();
var SizeYArray=new Array();
var SizeTArray=new Array();
var SizeXtext='';
var SizeYtext='';
var SizeTtext='';
var ZoomArray=new Array();
var Zoomtext='';

(function($){
     function onresize(){
          var Zoomtext=((window.outerWidth-16)/window.innerWidth);
          SizeXArray[SizeXArray.length]=$(window).width();
          SizeYArray[SizeYArray.length]=$(window).height();
          SizeTArray[SizeTArray.length]=Math.round(now()-start);
          ZoomArray[ZoomArray.length]=Zoomtext;
     };
     $(window).resize(onresize);
     onresize ();
})(jQuery);


///FUNCTION SEND / "NEXT" BUTTON

function send(){
y=new Date();

//RESPONSE TIME(2)

diff=Math.round(now()-start)

///CONNECTING VARIABLES WITH DATASET


//START TIME
```

```
MessStart=x.getTime();
var ZeitStart=document.forms[0].varZeitStart.value;
var ZeitStart_storage=eval("document.forms[0]."+ZeitStart);
ZeitStart_storage.value=MessStart;

//RESPONSE TIME
var ZeitDiff=document.forms[0].varZeitDiff.value;
var ZeitDiff_storage=eval("document.forms[0]."+ZeitDiff);
ZeitDiff_storage.value=diff;

//COMPASS
var CompassArrayOut=document.forms[0].varCompassArrayOut.value;
var CompassArray_storage=eval("document.forms[0]."+CompassArrayOut);
CompassArray_storage.value=CompassArray;

//COMPASS TIME
var CompassTArrayOut=document.forms[0].varCompassTArrayOut.value;
var CompassTArray_storage=eval("document.forms[0]."+CompassTArrayOut);
CompassTArray_storage.value=CompassTArray;

//CONNECTION TYPE
var ConnectionTypeOut=document.forms[0].varConnectionTypeOut.value;
var ConnectionType_storage=eval("document.forms[0]."+ConnectionTypeOut);
ConnectionType_storage.value=ConnectionType;

//DEVICE ORIENTATION AND ORIENTATION CHANGE
var ScreenOrientArrayOut=document.forms[0].varScreenOrientArrayOut.value;
var
ScreenOrientArrayOut_storage=eval("document.forms[0]."+ScreenOrientArrayOut
);
ScreenOrientArrayOut_storage.value=ScreenOrientArray;

//DEVICE ORIENTATION AND CHANGE TIME
var ScreenTArrayOut=document.forms[0].varScreenTArrayOut.value;
var ScreenTArrayOut_storage=eval("document.forms[0]."+ScreenTArrayOut);
ScreenTArrayOut_storage.value=ScreenTArray;

//GPS LATITUDE
var latitudeArrayOut=document.forms[0].varlatitudeArrayOut.value;
var latitudeArrayOut_storage=eval("document.forms[0]."+latitudeArrayOut);
latitudeArrayOut_storage.value=latitudeArray;

//GPS LONGITUDE
var longitudeArrayOut=document.forms[0].varlongitudeArrayOut.value;
var longitudeArrayOut_storage=eval("document.forms[0]."+longitudeArrayOut);
longitudeArrayOut_storage.value=longitudeArray;

//GPS ACCURACY
var accuracyArrayOut=document.forms[0].varaccuracyArrayOut.value;
var accuracyArrayOut_storage=eval("document.forms[0]."+accuracyArrayOut);
accuracyArrayOut_storage.value=accuracyArray;

//GPS ALTITUDE
var altituArrayOut=document.forms[0].varaltituArrayOut.value;
var altituArrayOut_storage=eval("document.forms[0]."+altituArrayOut);
altituArrayOut_storage.value=altituArray;

//GPS TIME
var GPStimeArrayOut=document.forms[0].varGPStimeArrayOut.value;
var GPStimeArrayOut_storage=eval("document.forms[0]."+GPStimeArrayOut);
GPStimeArrayOut_storage.value=GPStimeArray;
```

```
//GYROSCOPE ALPHA
var AlphaArray1=document.forms[0].varAlphaArray1.value;
var AlphaArray1_storage=eval("document.forms[0]."+AlphaArray1);
AlphaArray1_storage.value=AlphaArray;

//GYROSCOPE BETA
var BetaArray1=document.forms[0].varBetaArray1.value;
var BetaArray1_storage=eval("document.forms[0]."+BetaArray1);
BetaArray1_storage.value=BetaArray;

//GYROSCOPE GAMMA
var GammaArray1=document.forms[0].varGammaArray1.value;
var GammaArray1_storage=eval("document.forms[0]."+GammaArray1);
GammaArray1_storage.value=GammaArray;

//GYROSCOPE TIME
var GyroTimeArrayOut=document.forms[0].varGyroTimeArrayOut.value;
var GyroTimeArrayOut_storage=eval("document.forms[0]."+GyroTimeArrayOut);
GyroTimeArrayOut_storage.value=GyroTimeArray;

//KEY STROKES
var KeyCodeArrayOut=document.forms[0].varKeyCodeArrayOut.value;
var KeyCodeArrayOut_storage=eval("document.forms[0]."+KeyCodeArrayOut);
KeyCodeArrayOut_storage.value=KeyCodeArray;

//KEY STROKES TIME
var KeyTArrayOut=document.forms[0].varKeyTArrayOut.value;
var KeyTArrayOut_storage=eval("document.forms[0]."+KeyTArrayOut);
KeyTArrayOut_storage.value=KeyTArray;

//MOUSE CLICK AND FINGER TAB X
var ClickXArrayOut=document.forms[0].varClickXArrayOut.value;
var ClickXArrayOut_storage=eval("document.forms[0]."+ClickXArrayOut);
ClickXArrayOut_storage.value=ClickXArray;

//MOUSE CLICK AND FINGER TAB Y
var ClickYArrayOut=document.forms[0].varClickYArrayOut.value;
var ClickYArrayOut_storage=eval("document.forms[0]."+ClickYArrayOut);
ClickYArrayOut_storage.value=ClickYArray;

//MOUSE CLICK AND FINGER TAB TIME
var ClickTArrayOut=document.forms[0].varClickTArrayOut.value;
var ClickTArrayOut_storage=eval("document.forms[0]."+ClickTArrayOut);
ClickTArrayOut_storage.value=ClickTArray;

//MOUSE MOVEMENT X
var XArrayOut=document.forms[0].varXArrayOut.value;
var XArrayOut_storage=eval("document.forms[0]."+XArrayOut);
XArrayOut_storage.value=XArray;

//MOUSE MOVEMENT Y
var YArrayOut=document.forms[0].varYArrayOut.value;
var YArrayOut_storage=eval("document.forms[0]."+YArrayOut);
YArrayOut_storage.value=YArray;

//MOUSE MOVEMENT TIME
var TArrayOut=document.forms[0].varTArrayOut.value;
var TArrayOut_storage=eval("document.forms[0]."+TArrayOut);
TArrayOut_storage.value=TArray;

//SCREEN RESOLUTION X
var ScreenWArrayOut=document.forms[0].varScreenWArrayOut.value;
```

```
var ScreenWArrayOut_storage=eval("document.forms[0]."+ScreenWArrayOut);
ScreenWArrayOut_storage.value=ScreenWArray;

//SCREEN RESOLUTION Y
var ScreenHArrayOut=document.forms[0].varScreenHArrayOut.value;
var ScreenHArrayOut_storage=eval("document.forms[0]."+ScreenHArrayOut);
ScreenHArrayOut_storage.value=ScreenHArray;

//SCREEN RESOLUTION PIXEL RATIO
var ScreenRatioArrayOut=document.forms[0].varScreenRatioArrayOut.value;
var
ScreenRatioArrayOut_storage=eval("document.forms[0]."+ScreenRatioArrayOut);
ScreenRatioArrayOut_storage.value=ScreenRatioArray;

//SCROLLING (HORIZONTAL) X
var ScrollHXArrayOut=document.forms[0].varScrollHXArrayOut.value;
var ScrollHXArrayOut_storage=eval("document.forms[0]."+ScrollHXArrayOut);
ScrollHXArrayOut_storage.value=ScrollHXArray;

//SCROLLING (HORIZONTAL) TIME
var ScrollHTArrayOut=document.forms[0].varScrollHTArrayOut.value;
var ScrollHTArrayOut_storage=eval("document.forms[0]."+ScrollHTArrayOut);
ScrollHTArrayOut_storage.value=ScrollHTArray;

//SCROLLING (VERTICAL) Y
var ScrollVYArrayOut=document.forms[0].varScrollVYArrayOut.value;
var ScrollVYArrayOut_storage=eval("document.forms[0]."+ScrollVYArrayOut);
ScrollVYArrayOut_storage.value=ScrollVYArray;

//SCROLLING (VERTICAL) TIME
var ScrollVTArrayOut=document.forms[0].varScrollVTArrayOut.value;
var ScrollVTArrayOut_storage=eval("document.forms[0]."+ScrollVTArrayOut);
ScrollVTArrayOut_storage.value=ScrollVTArray;

//BASIC SURVEYFOCUS (SF)
var FocusArrayOut=document.forms[0].varFocusArrayOut.value;
var FocusArrayOut_storage=eval("document.forms[0]."+FocusArrayOut);
FocusArrayOut_storage.value=FocusArray;

//BASIC SUVEYFOCUS (SF) TIME
var FocusTArrayOut=document.forms[0].varFocusTArrayOut.value;
var FocusTArrayOut_storage=eval("document.forms[0]."+FocusTArrayOut);
FocusTArrayOut_storage.value=FocusTArray;

//MOBILE SURVEYFOCUS (SF)
var FocusArrayOut2=document.forms[0].varFocusArrayOut2.value;
var FocusArrayOut2_storage=eval("document.forms[0]."+FocusArrayOut2);
FocusArrayOut2_storage.value=FocusArray2;

//MOBILE SURVEYFOCUS (SF) TIME
var FocusTArrayOut2=document.forms[0].varFocusTArrayOut2.value;
var FocusTArrayOut2_storage=eval("document.forms[0]."+FocusTArrayOut2);
FocusTArrayOut2_storage.value=FocusTArray2;

//SURVEYMOTION (SMOTION) WITHOUT GRAVITY
var SM_ArrayOut=document.forms[0].varSM_ArrayOut.value;
var SM_ArrayOut_storage=eval("document.forms[0]."+SM_ArrayOut);
SM_ArrayOut_storage.value=SM_Array;

//SURVEYMOTION (SMOTION) WITHOUT GRAVITY TIME
var SM_T_ArrayOut=document.forms[0].varSM_T_ArrayOut.value;
var SM_T_ArrayOut_storage=eval("document.forms[0]."+SM_T_ArrayOut);
```

```
SM_T_ArrayOut_storage.value=SM_T_Array;

//SURVEYMOTION (SMOTION) WITH GRAVITY
var SM_G_ArrayOut=document.forms[0].varSM_G_ArrayOut.value;
var SM_G_ArrayOut_storage=eval("document.forms[0]."+SM_G_ArrayOut);
SM_G_ArrayOut_storage.value=SM_G_Array;

//SURVEYMOTION (SMOTION) WITH GRAVITY TIME
var SM_G_T_ArrayOut=document.forms[0].varSM_G_T_ArrayOut.value;
var SM_G_T_ArrayOut_storage=eval("document.forms[0]."+SM_G_T_ArrayOut);
SM_G_T_ArrayOut_storage.value=SM_G_T_Array;

//SWIPING
var SwipeArrayOut=document.forms[0].varSwipeArrayOut.value;
var SwipeArrayOut_storage=eval("document.forms[0]."+SwipeArrayOut);
SwipeArrayOut_storage.value=SwipeArray;

//SWIPING X
var SwipeTArrayOut=document.forms[0].varSwipeTArrayOut.value;
var SwipeTArrayOut_storage=eval("document.forms[0]."+SwipeTArrayOut);
SwipeTArrayOut_storage.value=SwipeTArray;

//SWIPING Y
var SwipeXArrayOut=document.forms[0].varSwipeXArrayOut.value;
var SwipeXArrayOut_storage=eval("document.forms[0]."+SwipeXArrayOut);
SwipeXArrayOut_storage.value=SwipeXArray;

//SWIPING TIME
var SwipeYArrayOut=document.forms[0].varSwipeYArrayOut.value;
var SwipeYArrayOut_storage=eval("document.forms[0]."+SwipeYArrayOut);
SwipeYArrayOut_storage.value=SwipeYArray;

//USER AGENT STRING
var UserAgentJSOut=document.forms[0].varUserAgentJSOut.value;
var UserAgentJSOut_storage=eval("document.forms[0]."+UserAgentJSOut);
UserAgentJSOut_storage.value=UserAgentJS;

//WINDOW SIZE X
var SizeXArrayOut=document.forms[0].varSizeXArrayOut.value;
var SizeXArrayOut_storage=eval("document.forms[0]."+SizeXArrayOut);
SizeXArrayOut_storage.value=SizeXArray;

//WINDOW SIZE Y
var SizeYArrayOut=document.forms[0].varSizeYArrayOut.value;
var SizeYArrayOut_storage=eval("document.forms[0]."+SizeYArrayOut);
SizeYArrayOut_storage.value=SizeYArray;

//WINDOW SIZE TIME
var SizeTArrayOut=document.forms[0].varSizeTArrayOut.value;
var SizeTArrayOut_storage=eval("document.forms[0]."+SizeTArrayOut);
SizeTArrayOut_storage.value=SizeTArray;

//ZOOM
var ZoomArrayOut=document.forms[0].varZoomArrayOut.value;
var ZoomArrayOut_storage=eval("document.forms[0]."+ZoomArrayOut);
ZoomArrayOut_storage.value=ZoomArray;

}

</script>
{/literal}
{* ECSP END *}
```

Appendix B

In the following, we share the ECSP HTML codes from 2016, 2018, and 2020. Before using ECSP, we highly recommend reading the chapters “ECSP and ethical considerations” and “ECSP disclaimer” above. We listed all codes in an alphabetical order, except for response and start times because they need to be collected first.

```
<!--EMBEDDED CLIENT SIDE PARADATA (Schlosser & Höhne, 2020) - HTML CODE-->


<!--START TIME-->
<input type=hidden name="varZeitStart" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--RESPONSE TIME-->
<input type=hidden name="varZeitDiff" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--COMPASS-->
<input type=hidden name="varCompassArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--COMPASS TIME-->
<input type=hidden name="varCompassTArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--CONNECTION TYPE-->
<input type=hidden name="varConnectionTypeOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--DEVICE ORIENTATION AND CHANGE-->
<input type=hidden name="varScreenOrientArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--DEVICE ORIENTATION AND CHANGE TIME-->
<input type=hidden name="varScreenTArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--GPS LATITUDE-->
<input type=hidden name="varlatitudeArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--GPS LONGITUDE-->
<input type=hidden name="varlongitudeArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--GPS ACCURACY-->
<input type=hidden name="varaccuracyArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--GPS ALTITUDE-->
<input type=hidden name="varaltituArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--GPS TIME-->
<input type=hidden name="varGPStimeArrayOut" value="v_XYZ">
```

```
<input type="hidden" name="v_XYZ" value="">

<!--GYROSCOPE ALPHA-->
<input type=hidden name="varAlphaArray1" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--GYROSCOPE BETA-->
<input type=hidden name="varBetaArray1" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--GYROSCOPE GAMMA-->
<input type=hidden name="varGammaArray1" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--GYROSCOPE TIME-->
<input type=hidden name="varGyroTimeArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--KEY STROKE-->
<input type=hidden name="varKeyCodeArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--KEY STROKE TIME-->
<input type=hidden name="varKeyTArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--MOUSE CLICK AND FINGER TAB X-->
<input type=hidden name="varClickXArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--MOUSE CLICK AND FINGER TAB Y-->
<input type=hidden name="varClickYArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--MOUSE CLICK AND FINGER TAB TIME-->
<input type=hidden name="varClickTArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--MOUSE MOVEMENT X-->
<input type=hidden name="varXArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--MOUSE MOVEMENT Y-->
<input type=hidden name="varYArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--MOUSE MOVEMENT TIME-->
<input type=hidden name="varTArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--SCREEN RESOLUTION X-->
<input type=hidden name="varScreenWArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--SCREEN RESOLUTION Y-->
<input type=hidden name="varScreenHArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--SCREEN RESOLUTION PIXEL RATIO-->
<input type=hidden name="varScreenRatioArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">
```

```
<!--SCROLLING (HORIZONTAL) X-->
<input type=hidden name="varScrollHXArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--SCROLLING (HORIZONTAL) TIME-->
<input type=hidden name="varScrollHTArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--SCROLLING (VERTICAL) Y-->
<input type=hidden name="varScrollVYArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--SCROLLING (VERTICAL) TIME-->
<input type=hidden name="varScrollVTArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--BASIC SURVEYFOCUS (SF)-->
<input type=hidden name="varFocusArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--BASIC SURVEYFOCUS (SF) TIME-->
<input type=hidden name="varFocusTArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--MOBILE SURVEYFOCUS (SF)-->
<input type=hidden name="varFocusArrayOut2" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--MOBILE SURVEYFOCUS (SF) TIME-->
<input type=hidden name="varFocusTArrayOut2" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--SURVEYMOTION (SMOTION) WITHOUT GRAVITY-->
<input type=hidden name="varSM_ArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--SURVEYMOTION (SMOTION) WITHOUT GRAVITY TIME-->
<input type=hidden name="varSM_T_ArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--SURVEYMOTION (SMOTION) WITH GRAVITY-->
<input type=hidden name="varSM_G_ArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--SURVEYMOTION (SMOTION) WITH GRAVITY TIME-->
<input type=hidden name="varSM_G_T_ArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--SWIPING-->
<input type=hidden name="varSwipeArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--SWIPING X-->
<input type=hidden name="varSwipeXArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--SWIPING Y-->
<input type=hidden name="varSwipeYArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--SWIPING TIME-->
<input type=hidden name="varSwipeTArrayOut" value="v_XYZ">
```

```
<input type="hidden" name="v_XYZ" value="">

<!--USER AGENT STRING-->
<input type=hidden name="varUserAgentJSOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--WINDOW SIZE X-->
<input type=hidden name="varSizeXArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--WINDOW SIZE Y-->
<input type=hidden name="varSizeYArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--WINDOW SIZE TIME-->
<input type=hidden name="varSizeTArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">

<!--ZOOM-->
<input type=hidden name="varZoomArrayOut" value="v_XYZ">
<input type="hidden" name="v_XYZ" value="">
```