

zTree

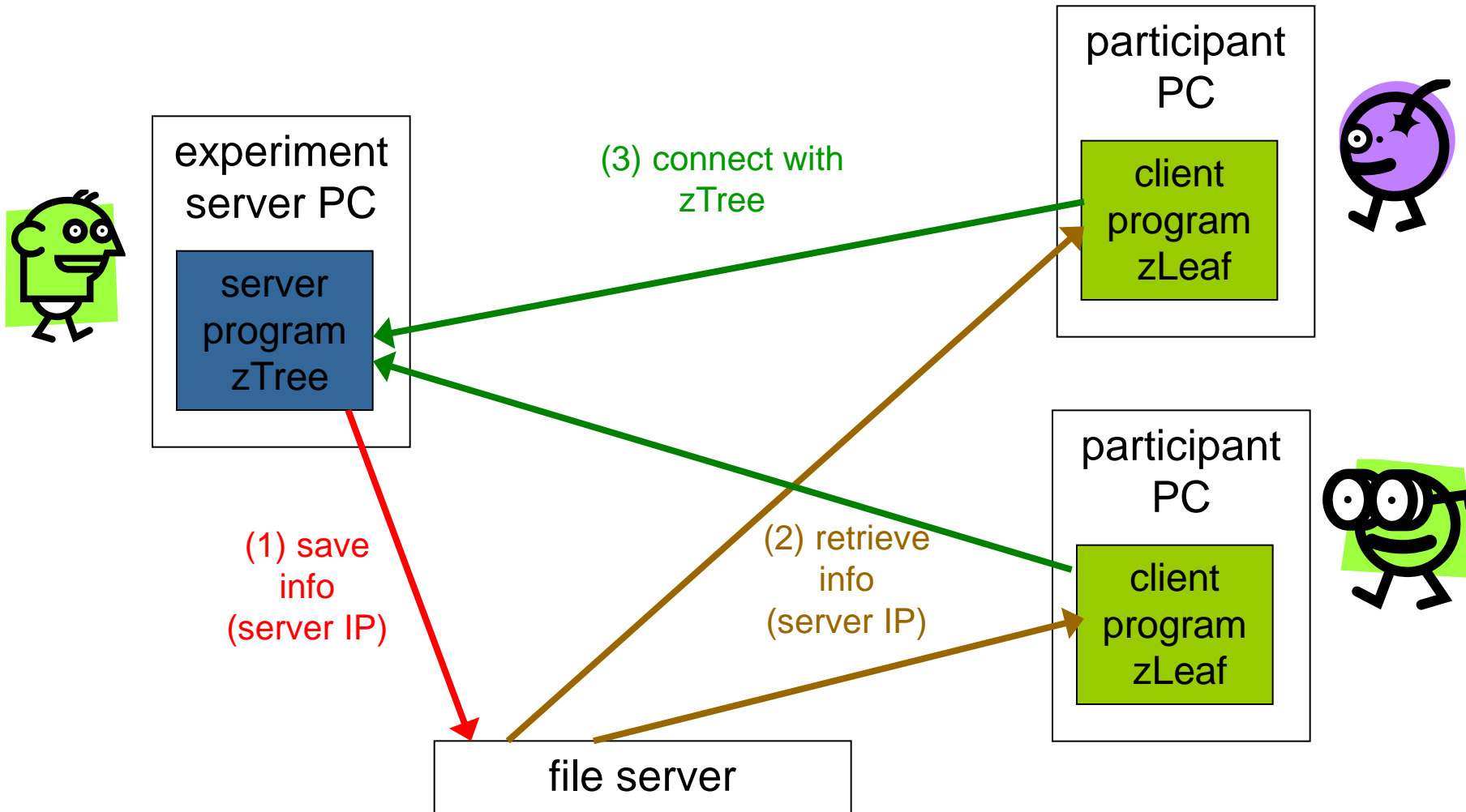
# zTree

---

- zTree – Zurich Toolbox for Readymade Economic Experiments
- by Urs Fischbacher
- <http://www.iew.unizh.ch/ztree/>
  
- Free, but not open source
- „Citeware“: you will have to cite Urs' journal article on the software in each publication of an experiment programmed with zTree
- ASBLab has signed an institutional license to this terms

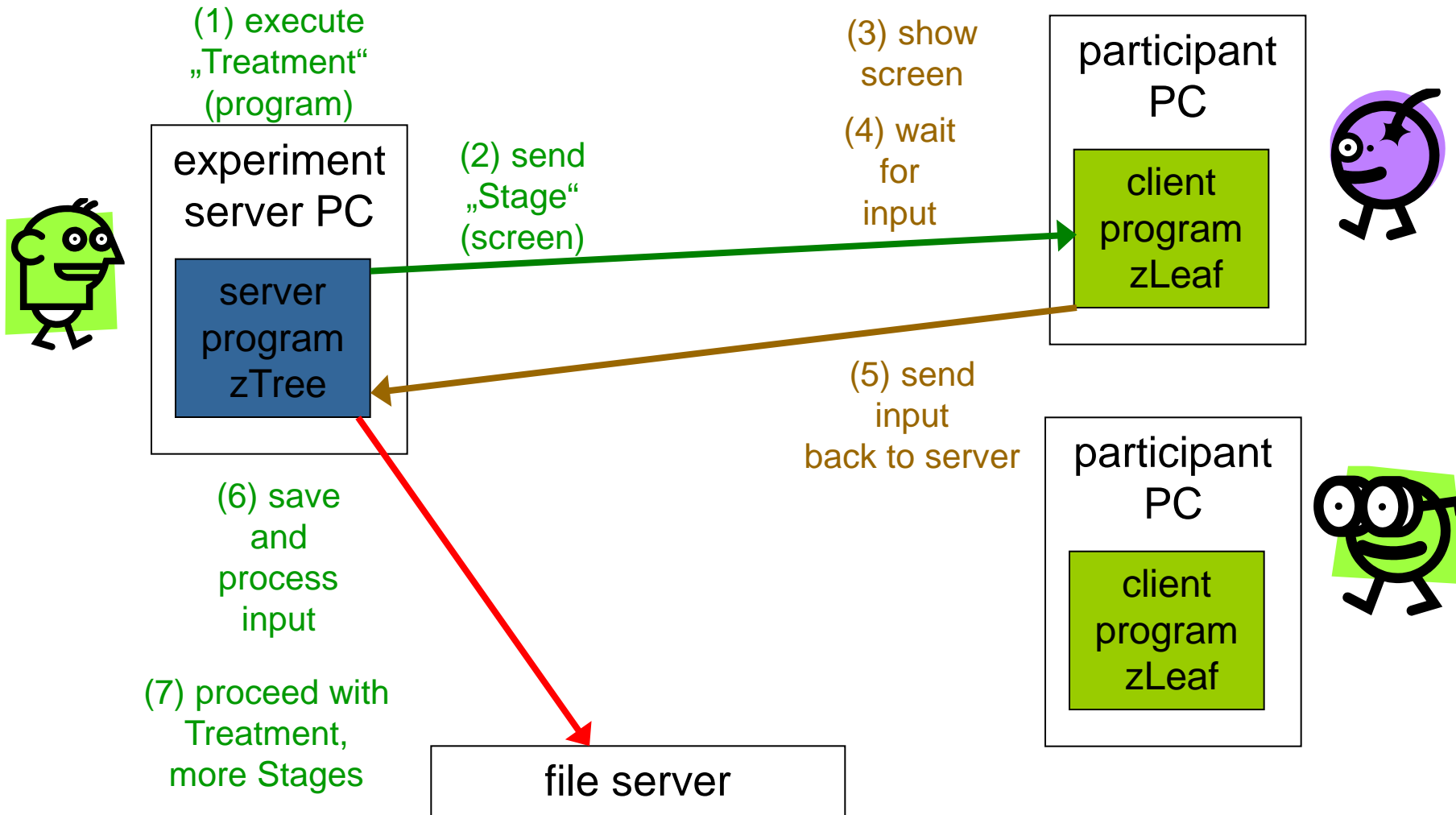
# Introduction zTree

- Client server architecture



# Introduction zTree

- Client server architecture



# Introduction zTree

## ■ The „Stage Tree“

The screenshot displays the zTree interface for a game simulation. The tree structure is as follows:

- Background
  - globals
  - subjects
  - Cooperation
  - summary
  - contracts
  - session
  - globals.do { ... }
  - subjects.do { ... }
    - // matching groups, 3 matching groups of 6 for 18 players
    - // subgroup = if (Subject < 7, 1, if (Subject < 13, 2, 3));
    - // times cooperated
    - Anzahl = if ( Period == 1, 0, OLDsubjects.find (same ( Subject), Anzahl));
    - // my Partner
    - // takes only the values from the first period in parameter table
    - partner = 0;
    - partner = if (Period == 1, subjects.find(same(Group) & not (same(Subject)), Subject), OLDsubjects.find(same(Subject), partner));
- Active screen
- Waitingsscreen

- Rematching =|=(30)A
- globals.do { ... }
- // set unlimited auction
- AuctionNoStop =1;
- subjects.do { ... }
- Active screen
  - List players: contracts( TRUE ), sorted by: SubNr
    - Player: OUT( SubNr )
    - Prev. Partner: OUT( prev\_partner )
    - Cooperated: OUT( cooperated )
  - Standard
    - Your are player: OUT( Subject )
    - Your matching status:: OUT( matched )
    - Your new Partner:: OUT( partner )
  - Send invitation: contracts( (got\_invitation == 0 & send\_invitation == 0 & matched == 0 & SubNr != :Subject) ), sorted by: SubNr
  - Answer invitation: contracts( SubNr == :Subject & got\_invitation == 1 & matched == 0 ), sorted by: gruppe
  - Waitingsscreen
- Rematching results =|=(-1)
- Decision =|=(30)
- Profit display =|=(30)

# Introduction zTree

- A screen

Woche \_\_\_\_\_ 10 von 12 \_\_\_\_\_ Verbleibende Zeit [sec]: 258

Aktie  A  B  C  D  E

Preis  Menge

**Geld:** 2000  
**Verfügbares Geld:** 1364

**Verkauf** **Kauf**

Sportler A	Sportler B	Sportler C	Sportler D	Sportler E
Stärke: 20	Stärke: 20	Stärke: 25	Stärke: 20	Stärke: 20
Anzahl: 20	Anzahl: 20	Anzahl: 20	Anzahl: 20	Anzahl: 20
Verfügbar: 20	Verfügbar: 20	Verfügbar: 5	Verfügbar: 7	Verfügbar: 20
bar:	bar:	bar:	ar:	bar:
<b>Buy</b>	<b>Buy</b>	<b>Buy</b>	<b>Buy</b>	<b>Buy</b>

**Eigene Gebote** **Löschen**

Preis	Menge	Preis	Menge	Preis	Menge	Preis	Menge	Preis	Menge	Aktie	K/V	Preis	Menge
25	5	21	15	43	5	34	13	23	8	A	Kauf	19	30
		20	5	40	10			21	10	B	Kauf	9	4
				39	5					B	Kauf	10	3
										C	Verkauf	39	5
										C	Verkauf	40	10
										D	Verkauf	34	13
20		20		20		20		20					
19	30	12	8	25	10	18	7	19	1				
18	7	10	3	22	4	10	4						
		9	4										
<b>Sell</b>	<b>Sell</b>	<b>Sell</b>	<b>Sell</b>	<b>Sell</b>	<b>Sell</b>	<b>Sell</b>	<b>Sell</b>	<b>Sell</b>	<b>Sell</b>				

# zTree Background and Tables

zTree - Untitled Treatment 1

File Edit Treatment Run Tools View ?

Untitled Treatment 1

- Background ← Properties of program
  - globals
  - subjects ← Definitions of tables
  - summary
  - contracts
  - session
- Active screen
  - Header ← Definitions of common elements on all screens (Stages)
  - Waitingsscreen
  - Text

Bitte warten Sie bis das Experiment weiter geht.

General Parameters

Number of subjects

Number of groups

# practice periods

# paying periods

Exch. rate [Fr./ECU]

Lump sum payment [ECU]

Show up fee [Fr.]

OK

Cancel

Bankruptcy rules...

Header Box

Name   With frame

Width [p/%)

Height [p/%)

Distance to the margin [p/%)

Adjustment of the remaining box

left  top  right

bottom

OK

Cancel

Display condition

Show current period number

Show total number of periods

Name of "Period"

Term for "out of"

Prefix for trial periods

Display time

Term for "Remaining time"

Term for "Please reach a decision"

Table

Name

OK

Cancel

Lifetime

Period

Treatment

Session

Program execution

When first subject enters stage

Always when a subject enters stage

When last subject enters stage

# zTree Parameter Table

sess2\_solidarity\_noinfo\_asym\_27\_1.ztt:2

	S 1	S 2	S 3	S 4	S 5	S 6	S 7	S 8	S 9	S 10	S 11	S 12	S 13	S 14	S 15	S 16	S 17	S 18	S 19	S 20	S 21	S 22	S 23	S 24	S 25	S 26	S 27
1	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
2	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
3	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
4	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
5	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
6	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
7	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
8	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
9	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
10	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
11	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
12	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
13	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
14	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
15	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
16	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
17	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
18	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
19	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
20	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
21	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
22	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3
23	2	3	1	3	2	3	2	1	2	1	2	1	3	1	2	3	2	3	1	2	1	3	1	3	1	2	3

**Set variable values in table „subjects“ for periods, subjects, or a specific subject in a specific period**

**Important: Group variable**

**Specific Parameter**

Subject S 13      Period 3

Group

Name

Program  
 Group = 3;  
 drop\_out\_round = 0;  
 candidate = 2;  
 drop\_out\_threshold = 0.893250002;

OK      Cancel



# zTree – The „Stage“

**example.ztt**

- ExperimentResults =| = (final\_screen\_time)N
  - globals.do { ... } ← **globals – program – executed 1x**
    - sum\_contributions = subjects.sum(contribution);
  - subjects.do { ... } ← **subjects – program – executed for all subjects one by one**
    - contribution\_partner = subjects.find(Group == :Group & Subject != :Subject, contribution);  
: – **the scope operator: access to higher level tables**
    - Profit = endowment + contribution\_partner;
    - Participate = if (Period==NumPeriods,1,0); ← **Participate – sets whether a Subject sees a screen or not**
  - Active screen
    - Standard ← **box**
      - Das Experiment ist jetzt zu Ende.
      - <> Sie haben in diesem Experiment insgesamt <overall\_profit | 1> ECU verdient. ← **items in box**
      - <> Dieses entspricht einem Betrag von <money\_profit | 0.01> EURO.
      - 
      - Bitte klicken Sie den OK-Button.
      - OK ← **button**
    - Waiting screen
      - FinalResults
        - Das Experiment ist jetzt zu Ende.

# Documentation

---

- Current version: 3.3.8
- Tutorial (v 2.2)
  - Introduction to programming in zTree
  - Many examples and tips & tricks
- Reference Manual (v 2.2)
  - Complete reference
- **Wiki**
- zTree mailing list