Appendix – **Common Contracts**

A.1 Contracts

There are several contracts that are shared by all case studies:

- `contracts/common/base.sol`
  Contains the abstract choreography contract which all choreographies inherit.
- `contracts/common/interfaces.sol`
  Contains all interfaces used to communicate between and with the choreographies.
- `contracts/common/participants.sol`
  Contains the participants registry contract.

Additionally, for each case study, we provide

- in `models/*.bpmn2`
  the BPMN2 XML model file used to generate the smart contract code as well as
- in `contracts/*/`
  all Solidity smart contracts generated by the proof-of-concept implementation.

The latter are numbered by their appearance in the model file.
Each smart contract corresponds to the root choreography or a sub/call choreography.

A.2 Rental Agreement

A.2.1 Models

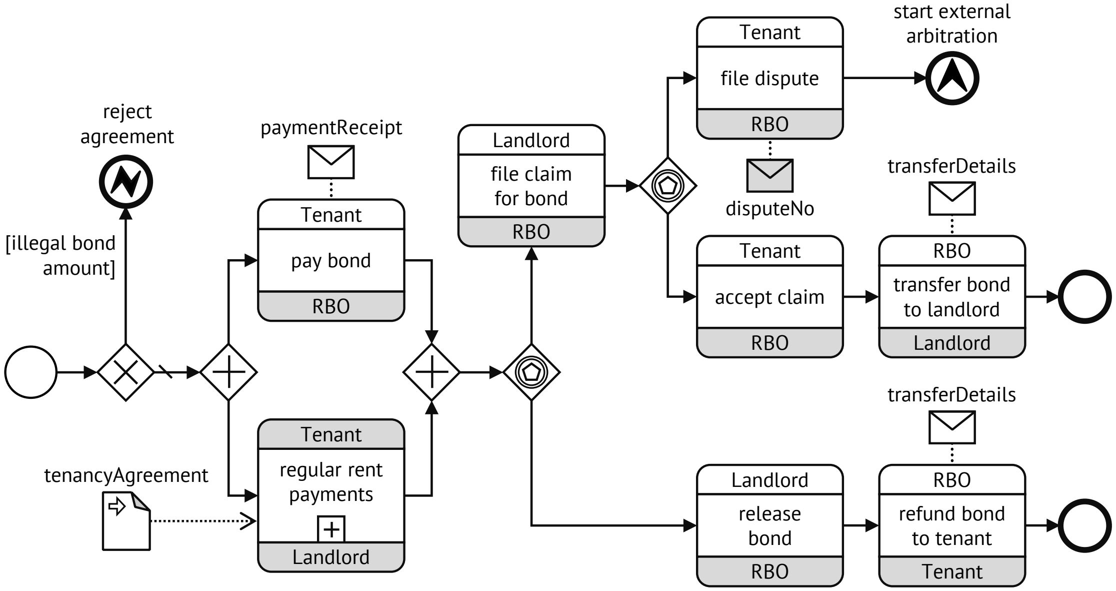

Fig. A.1. Top-level root choreography of the rental agreement case study

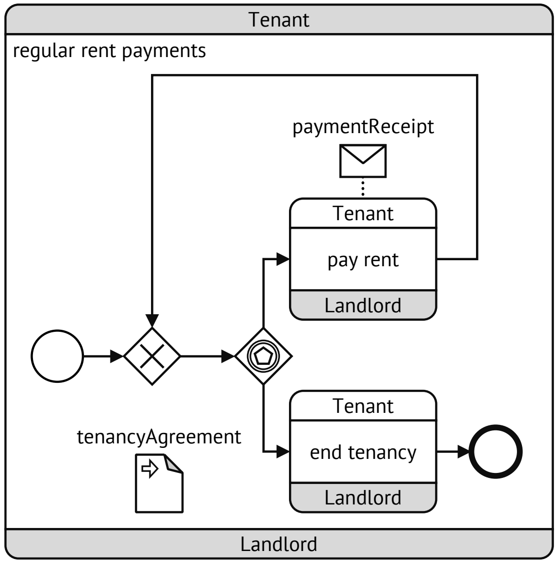

Fig. A.2. Expanded sub-choreography handling the regular rent payments of the rental agreement case study

A.2.2 Refinement

Data Structures (Messages and Data Objects)

tenancyAgreement

```
[["uint16","bond"],["uint16","weeklyRent"]]
```

paymentReceipt

```
[["uint32","receiptID"]]
```

disputeNo

```
[["uint32","disputeNo"]]
```

transferDetails

```
[["int32","timestamp"],["uint32","transferID"]]
```

Guard Expressions

[illegal bond amount]

```
tenancyAgreement_bond > 4 * tenancyAgreement_weeklyRent
```

A.2.3 Gas Costs

Rental Agreement
The landlord files a claim for the bond which the tenant disputes.

	action	**participant**	**gas**	**comment**
factories	*deploy factory root_0*	any	1,195,765	factory for "regular rent payments"
			1,195,765	
deployment	*deploy participants container*	any	285,681	deploy the participants container
	deploy root choreography	any	1,737,024	deploy a new instance of the root choreography [input (400, 250)]
			2,022,705	
transactions	*init root*	any	868,760	(includes 1 sub-choreography deployment worth ~778,929 gas)
	pay bond (ID 50)	Tenant	55,743	
	init root_0	any	40,832	
	pay rent (ID 42)	Tenant	48,902	
	end tenancy	Tenant	50,429	
	file claim for bond	Landlord	47,098	
	file dispute	Tenant	38,691	
	file dispute reply (disp. no. 13)	RBO	63,895	
			1,214,350	
			478,160	*average per transaction*

A.3 Grain Delivery

A.3.1 Models

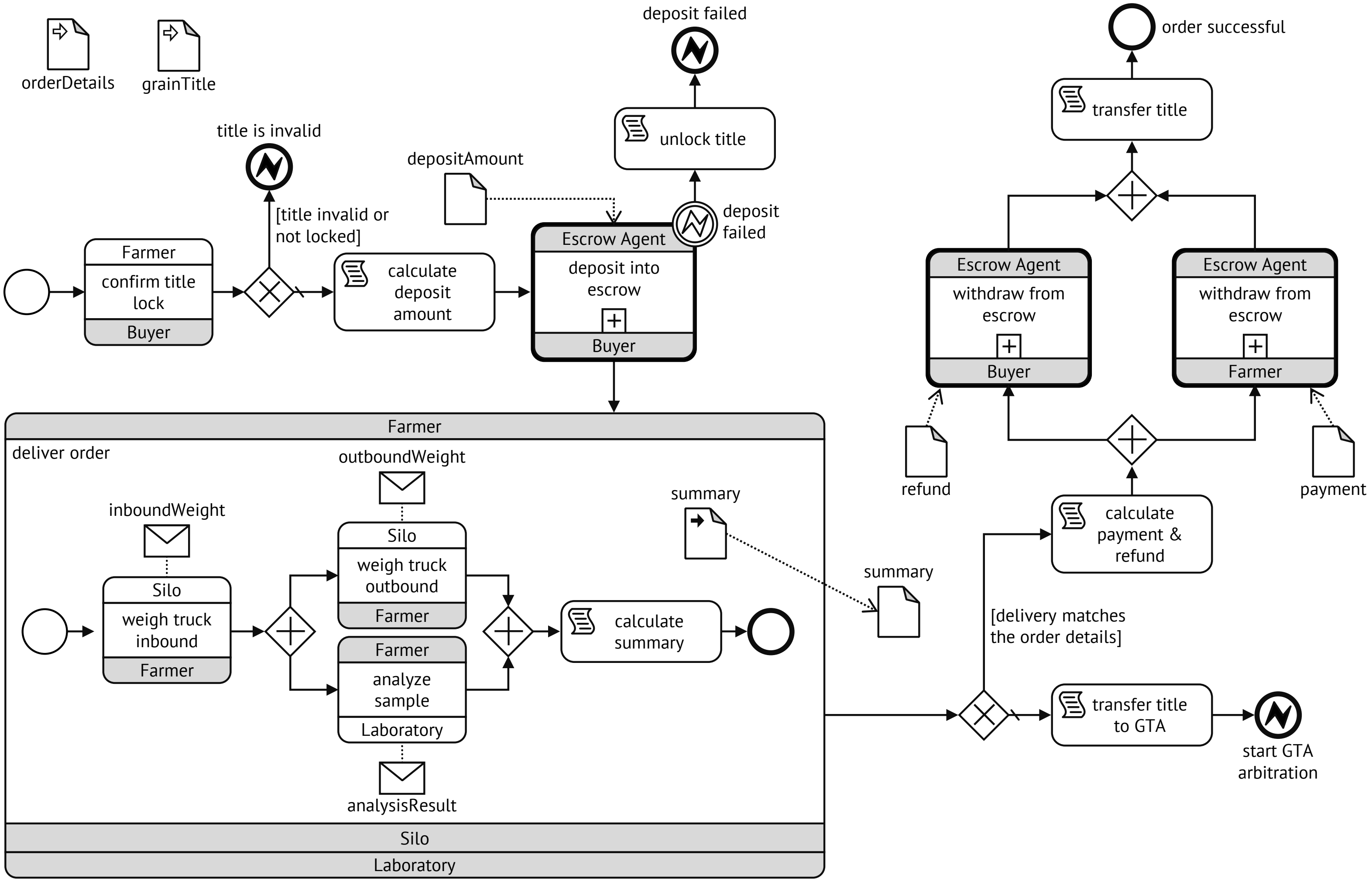

Fig. A.3. Top-level root diagram of the grain delivery case study

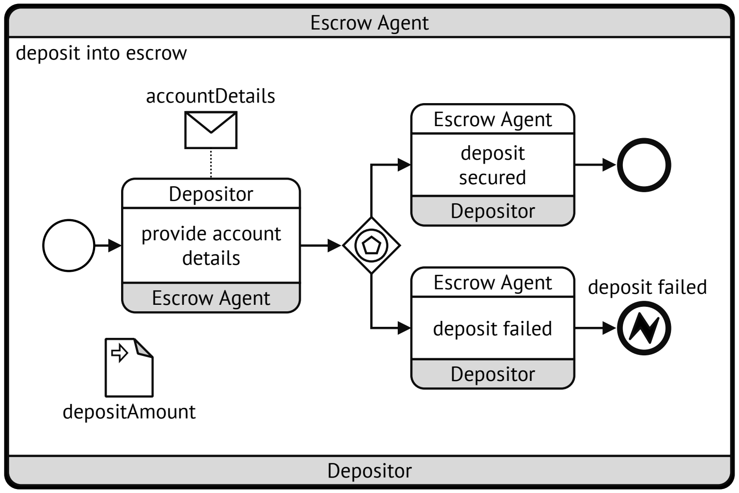

Fig. A.4. Expanded call choreography handling the escrow deposit of the grain delivery case study

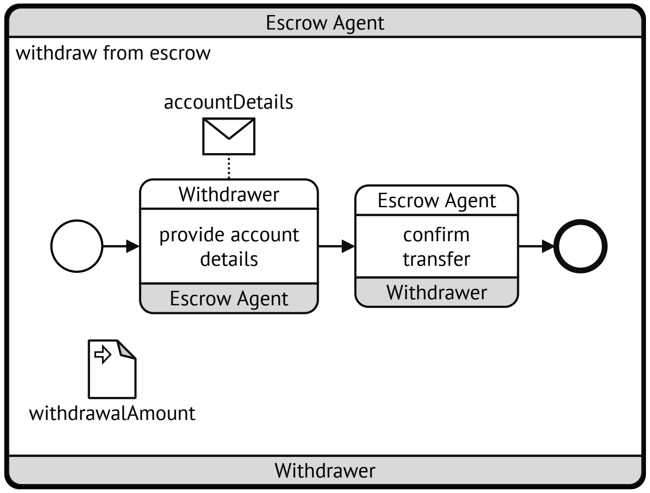

Fig. A.5. Expanded call choreography handling the escrow withdrawal of the grain delivery case study

A.3.2 Refinement

Data Structures (Messages and Data Objects)

orderDetails

```
[["uint8","grade"],["uint16","tonnes"],
 ["uint16","tolerance"],["uint16","price"]]
```

grainTitle

```
[["address","addr"]]
```

depositAmount, refund, payment, withdrawalAmount

```
[["uint16","amount"]]
```

summary

```
[["uint8","grade"],["uint16","tonnes"]]
```

analysisResult

```
[["uint8","grade"]]
```

inboundWeight, outboundWeight

```
[["uint16","tonnes"]]
```

accountDetails

```
[["uint16","bsb"],["uint32","account"]]
```

Scripts

calculate deposit amount

```
depositAmount_amount = (orderDetails_tonnes +
  orderDetails_tolerance) * orderDetails_price;
```

unlock title

```
grainTitle_addr.call(bytes4(keccak256("unlock()")));
```

calculate summary

```
summary_grade = analysisResult_grade;
summary_tonnes =
  inboundWeight_tonnes - outboundWeight_tonnes;
```

transfer title to GTA (replace 0x0 with actual GTA address)

```
grainTitle_addr.call(bytes4(keccak256("assign(address)")), 0x0);
```

calculate payment and refund

```
payment_amount = summary_tonnes * orderDetails_price;
refund_amount = depositAmount_amount - payment_amount;
```

transfer title to buyer

```
grainTitle_addr.call(
  bytes4(keccak256("assign(address)")), participants.get(1)
);
```

Guard Expressions

[title invalid or not locked]

```
!grainTitle_addr.call(bytes4(keccak256("amTrustee()")))
```

[delivery matches the order details]

```
(summary_tonnes >= orderDetails_tonnes - orderDetails_tolerance) &&
(summary_tonnes <= orderDetails_tonnes + orderDetails_tolerance) &&
(summary_grade >= orderDetails_grade)
```

A.3.3 Gas Costs

Grain Delivery

Grain is successfully delivered conforming to the contractual agreement.

	action	participant	gas	comment
factories	*deploy factory root_0*	any	1,278,011	factory for "deposit into escrow"
	deploy factory root_1	any	1,277,953	factory for "deliver order"
	deploy factory root_2	any	1,188,791	factory for "withdraw from escrow" to Buyer
	deploy factory root_3	any	1,189,059	factory for "withdraw from escrow" to Farmer
			4,933,814	
deployment	*deploy participants container*	any	285,681	deploy the participants container
	deploy root choreography	any	1,906,166	deploy a new instance of the root choreography
			2,191,847	
transactions	*init root*	any	41,261	
	confirm title lock	Farmer	947,513	(includes 1 call choreography deployment worth ~848,015 gas)
	init root_0	any	38,093	
	provide account details	Buyer	47,727	
	deposit secured	Escrow Agent	947,002	(includes 1 sub-choreography deployment worth ~853,773 gas)
	init root_1	any	38,813	
	weigh truck inbound	Silo	48,864	
	weigh truck outbound	Silo	47,145	
	analyze sample	Laboratory	1,733,746	(includes 2 call choreography deployments worth ~1,556,544 gas)
	init root_2	any	37,654	
	provide account details	Buyer	45,996	
	confirm transfer	Escrow Agent	50,793	
	init root_3	any	37,654	
	provide account details	Farmer	46,002	
	confirm transfer	Escrow Agent	67,258	
			4,175,521	
			278,368	*average per transaction*

A.4 Interline Agreement

A.4.1 Models

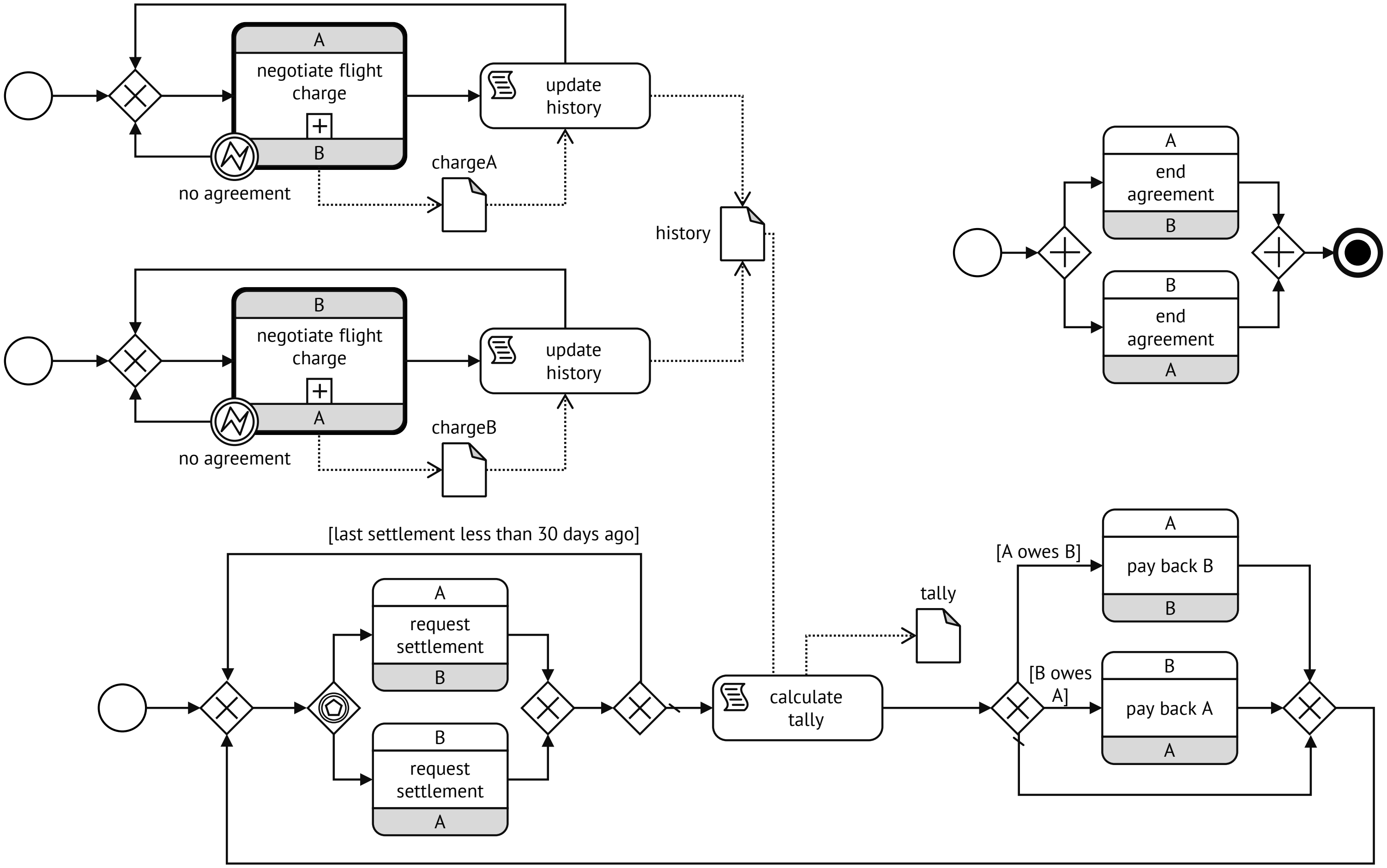

Fig. A.6. Top-level root choreography modeling the interline agreement case study

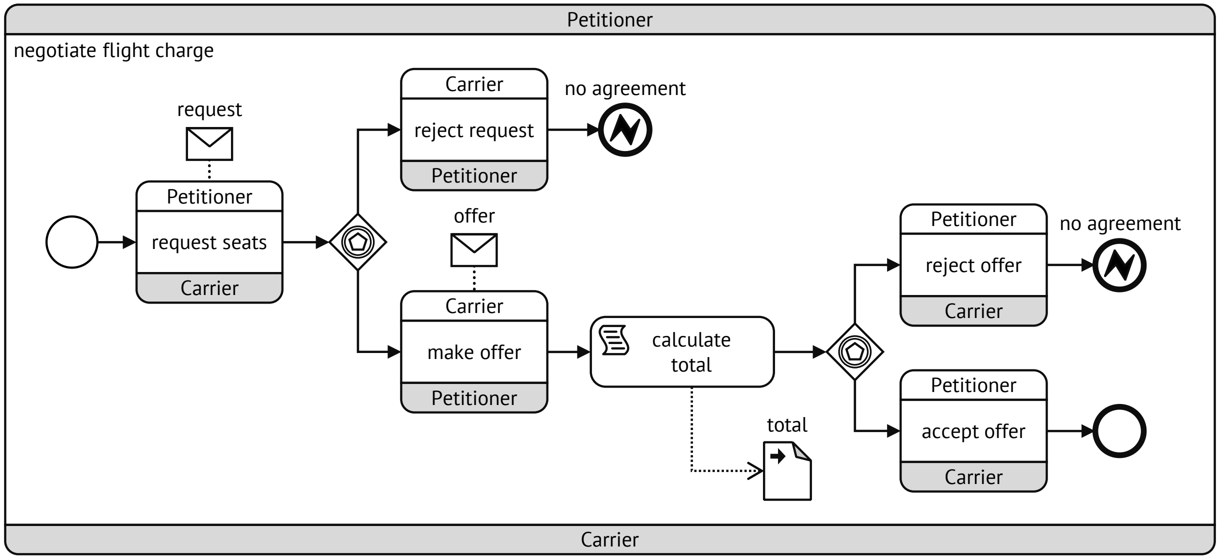

Fig. A.7. Expanded call choreography of the flight charge negotiation of the interline agreement case study

A.4.2 Refinement

Data Structures (Messages and Data Objects)

chargeA, chargeB, total

```
[["uint32","charge"]]
```

history

```
[["uint32","debtA"],["uint32","debtB"],["uint64","lastSettlement"]]
```

tally

```
[["int40","tally"]]
```

request

```
[["uint8","noOfSeats"]]
```

offer

```
[["uint32","pricePerSeat"]]
```

Scripts

update history (for A)

```
history_debtA += chargeA_charge;
```

update history (for B)

```
history_debtB += chargeB_charge;
```

calculate tally

```
tally_tally = history_debtA - history_debtB;
history_debtA = 0;
history_debtB = 0;
history_lastSettlement = uint64(now);
```

calculate total

```
total_charge = request_noOfSeats * offer_pricePerSeat;
```

Guard Expressions

[last settlement less than 30 days]

```
now < history_lastSettlement + 30 days
```

[A owes B]

```
tally_tally > 0
```

[B owes A]

```
tally_tally < 0
```

A.4.3 Gas Costs

Interline Agreement
Gas cost if both airlines successfully negotiate a flight charge, settle and then end the agreement.

	action	**participant**	**gas**	**comment**
factories	*deploy factory root_0*	any	1,413,128	factory for "negotiate flight charge" from A to B
	deploy factory root_1	any	1,412,196	factory for "negotiate flight charge" from B to A
			2,825,324	
deployments	*deploy participants container*	any	285,681	deploy the participants container
	deploy root choreography	any	1,807,957	deploy a new instance of the root choreography
			2,093,638	
transactions	*init root*	any	2,115,911	(includes 2 call choreography deployments worth ~1,911,102 gas)
	init root_0	any	39,030	
	request seats (10)	A	49,408	
	make offer (80)	B	57,010	
	accept offer	A	1,071,406	(includes 1 call choreography deployment worth ~955,851 gas)
	init root_1	any	39,030	
	request seats (2)	B	49,399	
	make offer (200)	A	57,001	
	accept offer	B	1,070,962	(includes 1 call choreography deployment worth ~955,251 gas)
	request settlement	A	74,269	
	pay back A to B	A	64,069	
	end agreement	A	46,905	
	end agreement	B	59,899	
			4,794,299	
			368,792	*average per transaction*