

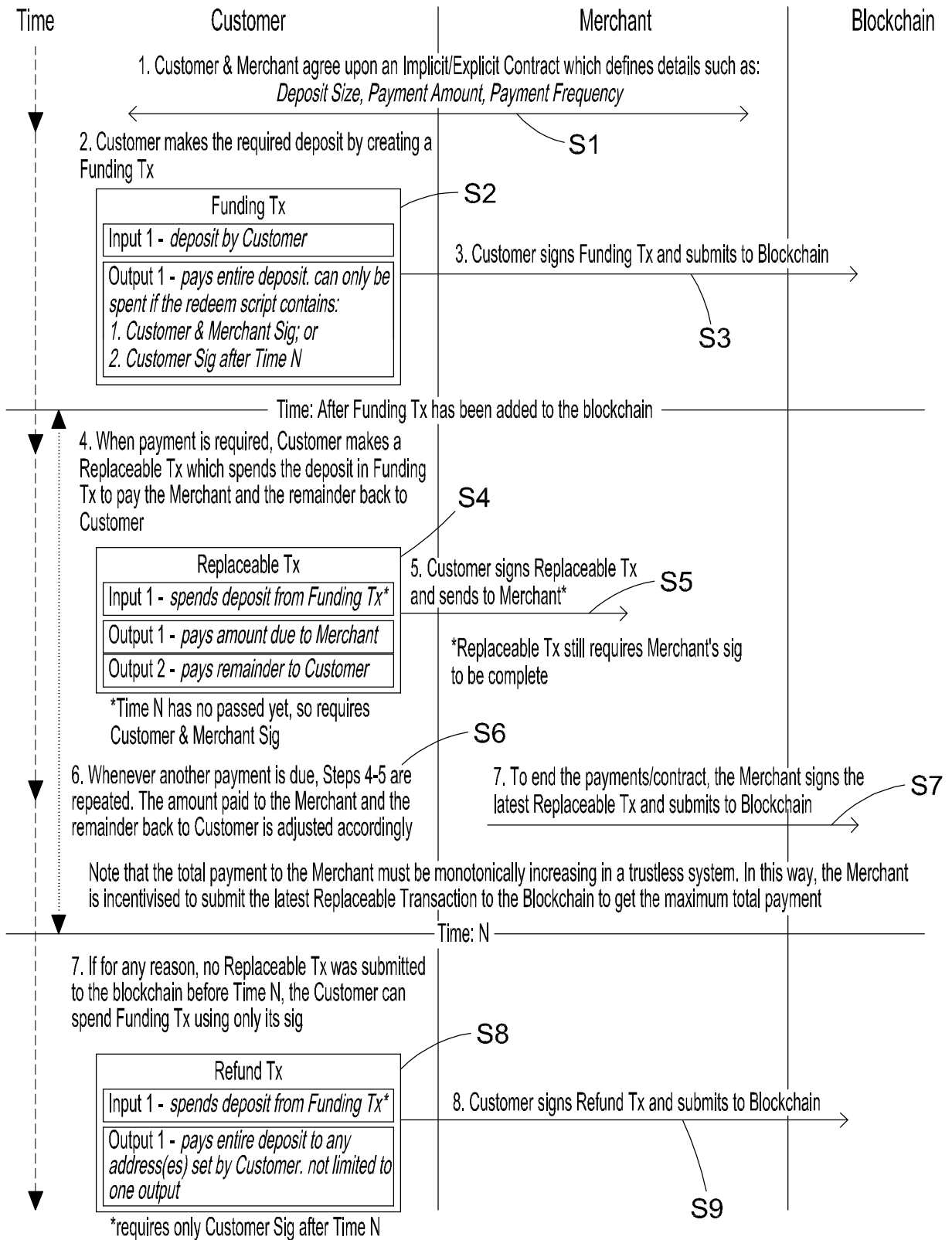


Fig. 1

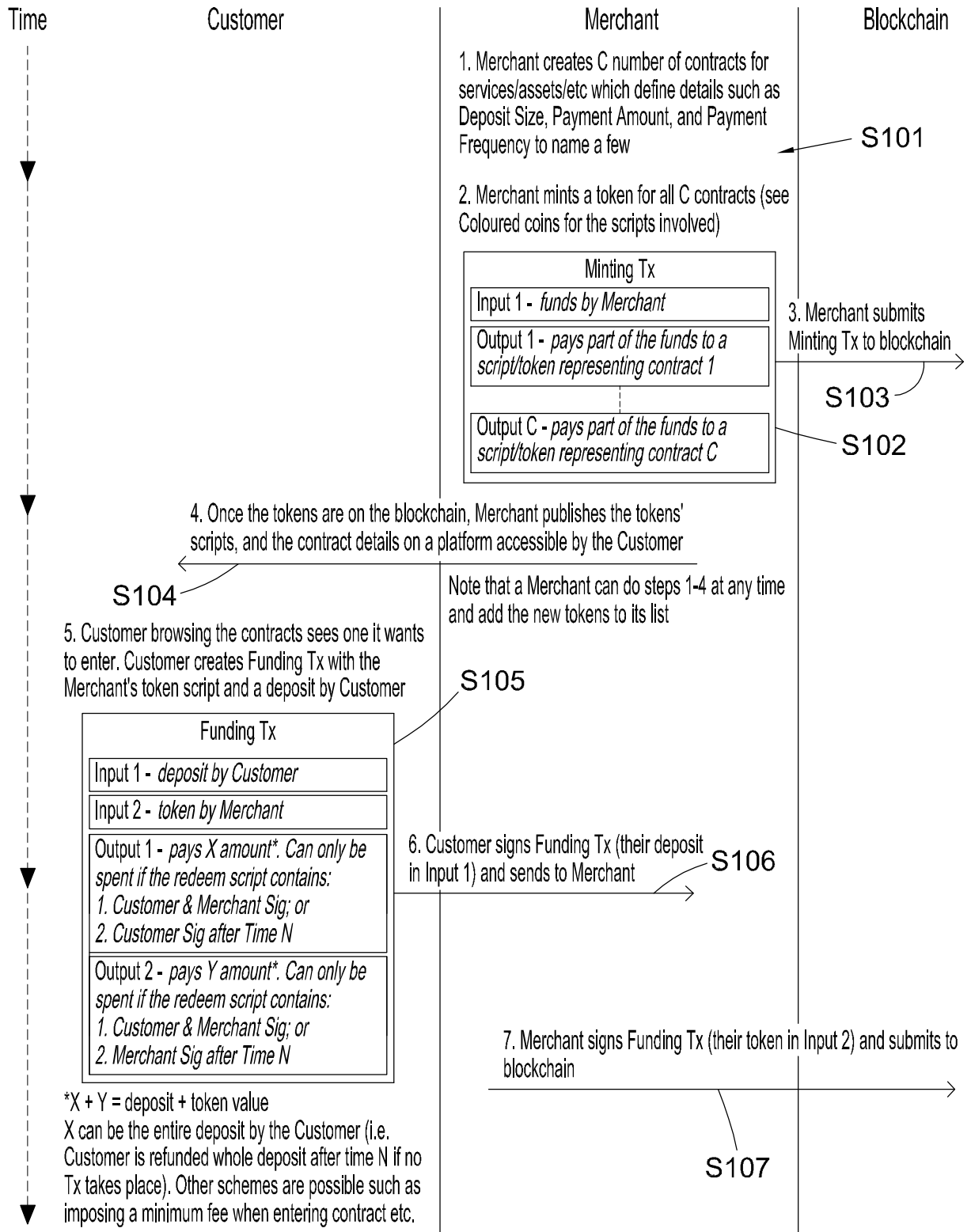


Fig. 2

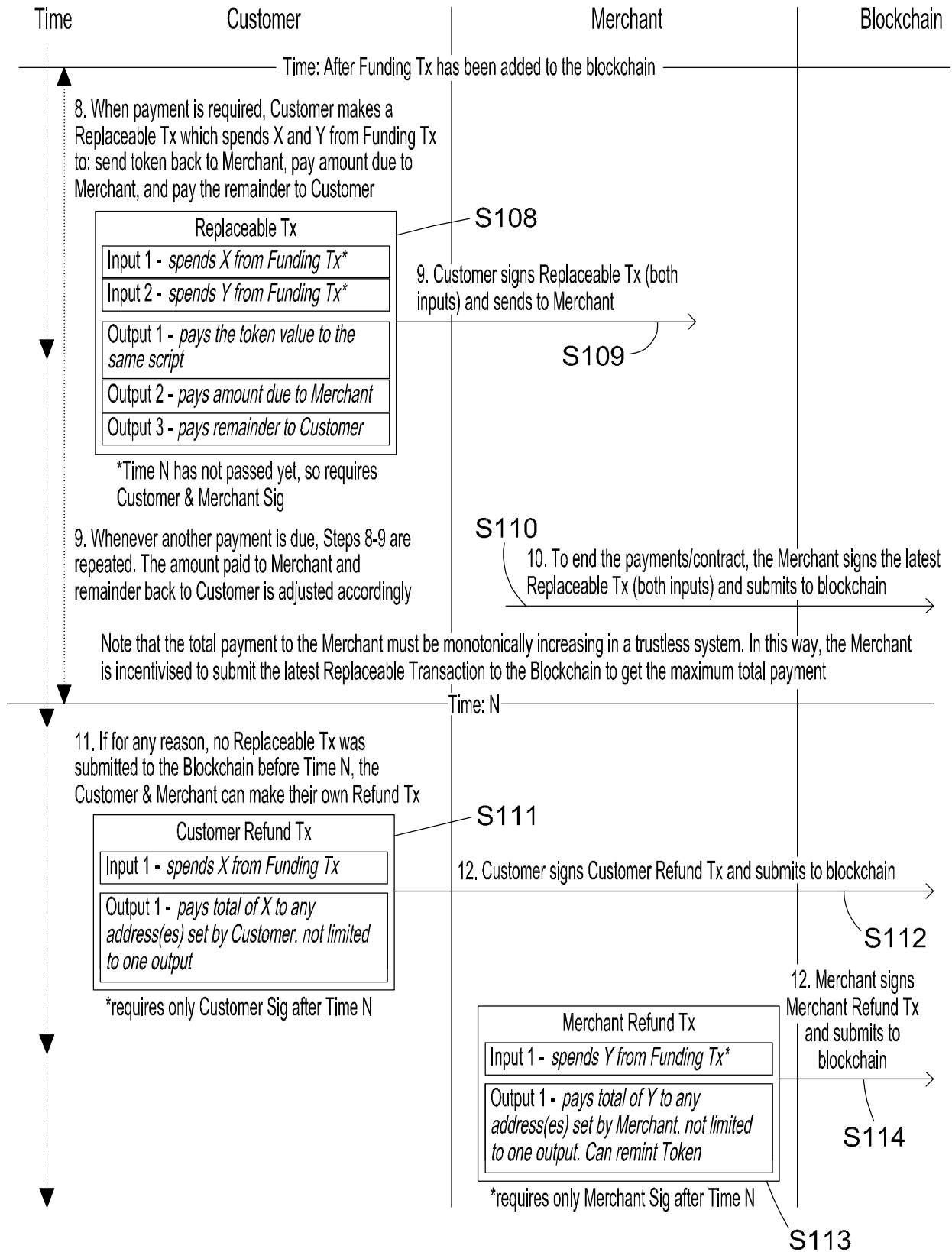


Fig. 2 - continued

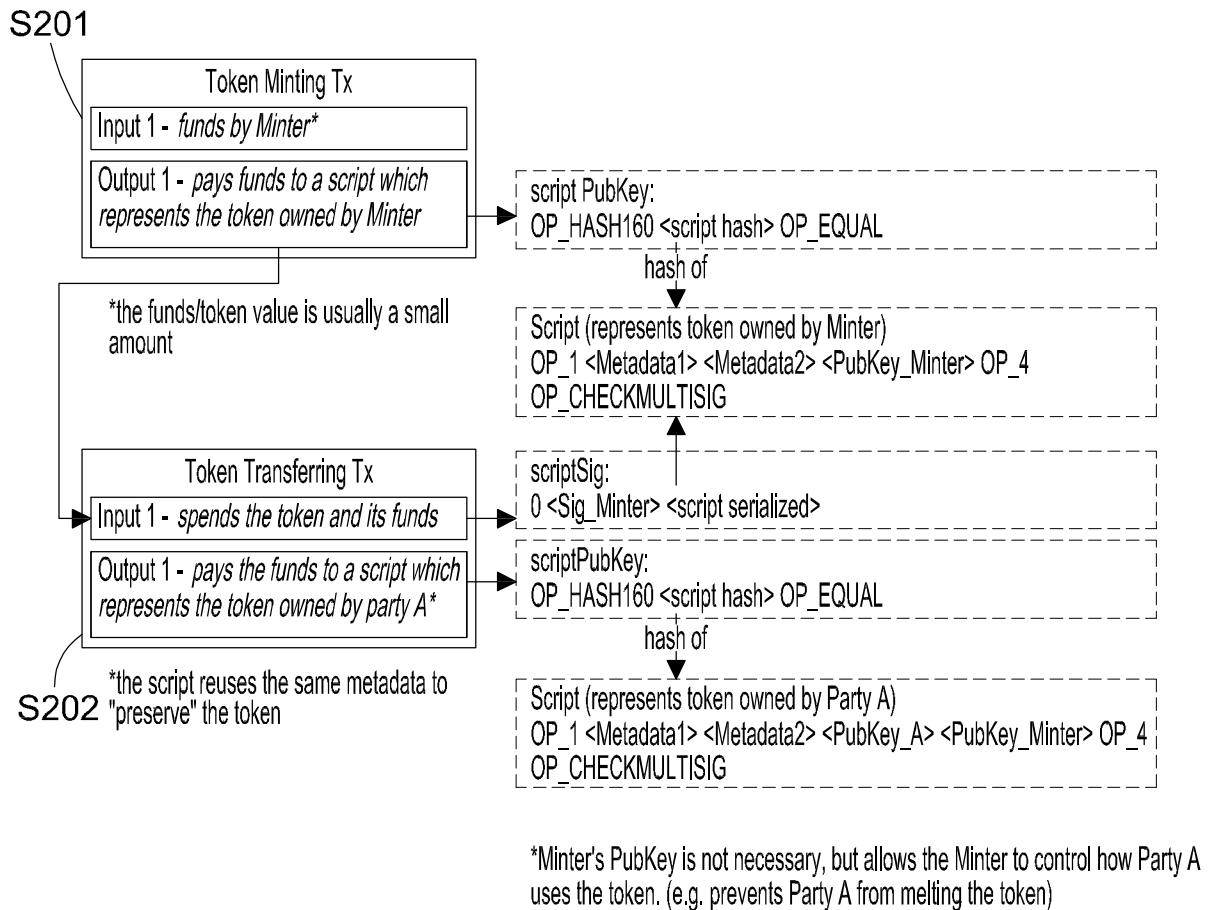


Fig. 3

Minting Transaction Details

Minting Transaction	
MINTING_TX	
1	
<Tx containing Funds by Merchant, assume 1,000 Satoshis>	
<Tx Index of Funds>	
<Script required to unlock Funds>	
<ScriptSignatureFlags>	
0xFFFFFFFF	
1	
1,000	
OP_HASH160 <script_hash_token> OP_EQUAL	
0	

Fig. 4A

- Input 1 - funds by Merchant to mint the token
 - Funds amount and unlocking script is arbitrary
- Output 1 - pays funds to <script_hash_token> which is Hash160 of the script:

Script	OP_1 <Metadata1> <Metadata2> <PubKey_Merchant> OP_3 OP_CHECKMULTISIG where metadata1 & metadata2 are pretend compressed public keys containing information about the contract
--------	--

Fig. 4B

Funding Transaction Details

Funding Transaction	
FUNDING_TX	
2	
<Tx containing Deposit by Customer, assume 500,000 Satoshis>	
<Tx Index of Deposit>	
<Script required to unlock Deposit>	
<ScriptSignatureFlags>	
0xFFFFFFFF	
MINTING_TX	
IDX_001	
<Sig Merchant> <script_serialized_token>	
<ScriptSignatureFlags>	
0xFFFFFFFF	
2	
450,000	
OP_HASH160 <script_hash_customer> OP_EQUAL	
51,000	
OP_HASH160 <script_hash_merchant> OP_EQUAL	
0	

Fig. 5A

- Input 1 - deposit by the Customer as required by contract
 - The deposit amount and unlocking method is arbitrary
- Input 2 - token by Merchant, Customer copies this from a platform where the Merchant publishes the contract and token script. <script_serialized_token> is serialized token script:

Script	OP_1 <Metadata1> <Metadata2> <PubKey_Merchant> OP_3 OP_CHECKMULTISIG where metadata1 & metadata2 contain information about the contract
--------	---

- Output 1 - pays X amount to <script_hash_customer> which is a Hash160 of:

Script	OP_IF <now + time N> OP_CHECKLOCKTIMEVERIFY OP_DROP <PubKey_Customer> OP_CHECKSIGVERIFY OP_ELSE <PubKey_Customer> <PubKey_Merchant> OP_2 OP_CHECKMULTISIGVERIFY OP_ENDIF
--------	---

- <script_hash_customer> can be unlocked by:
 - OP_1 <Sig_Customer> <serialized script customer> with Locktime N; or
 - OP_0 <Sig_Customer> <Sig Merchant> <serialized script customer>
- X amount paid (450,000 here) is adjustable depending on contract terms. i.e. how much is refunded to Customer if no Replaceable Transaction is submitted to the blockchain before time N

- Output 2 - pays Y amount to <script_hash_merchant> which is a Hash160 of:

Script	OP_IF <now + time N> OP_CHECKLOCKTIMEVERIFY OP_DROP <PubKey_Merchant> OP_CHECKSIGVERIFY OP_ELSE <PubKey_Customer> <PubKey_Merchant> OP_2 OP_CHECKMULTISIGVERIFY OP_ENDIF
--------	---

- <script_hash_merchant> can be unlocked by:
 - OP_1 <Sig_Merchant> <serialized script merchant> with Locktime N; or
 - OP_0 <Sig_Customer> <Sig Merchant> <serialized script merchant>
- Y amount paid (50,000 here) is adjustable depending on contract terms. i.e. how much is paid to Merchant if no Replaceable Transaction is submitted to the blockchain before time N

Fig. 5B

Replaceable Transaction Details

Replaceable Transaction	
REPLACEABLE_TX	
2	
FUNDING_TX	
IDX_001	
OP_0 <Sig_Customer> <Sig_Merchant> <serialized script customer>	
<ScriptSignatureFlags>	
0xFFFFFFFF	
FUNDING_TX	
IDX_002	
OP_0 <Sig_Customer> <Sig_Merchant> <serialized script merchant>	
<ScriptSignatureFlags>	
0xFFFFFFFF	
3	
1,000	
OP_HASH160 <Script_hash_token> OP_EQUAL	
10,000	
OP_DUP OP_HASH160 <Address_Merchant> OP_EQUALVERIFY	
OP_CHECKSIG	
490,000	
OP_DUP OP_HASH160 <Address_Customer> OP_EQUALVERIFY	
OP_CHECKSIG	
0	

Fig. 6A

- Input 1 - spends Output 1 of Funding Tx using Customer & Merchant sig to unlock <serialized script customer>:

Script	OP_IF <now + time N> OP_CHECKLOCKTIMEVERIFY OP_DROP <PubKey_Customer> OP_CHECKSIGVERIFY OP_ELSE <PubKey_Customer> <PubKey_Merchant> OP_2 OP_CHECKMULTISIGVERIFY OP_ENDIF
--------	---

- Input 2 - spends Output 2 of Funding Tx using Customer & Merchant sig to unlock <serialized script merchant>:

Script	OP_IF <now + time N> OP_CHECKLOCKTIMEVERIFY OP_DROP <PubKey_Merchant> OP_CHECKSIGVERIFY OP_ELSE <PubKey_Customer> <PubKey_Merchant> OP_2 OP_CHECKMULTISIGVERIFY OP_ENDIF
--------	---

- Output 1 - pays funds to <script_hash_token> which is Hash160 of the script:

Script	OP_1 <Metadata1> <Metadata2> <PubKey_Merchant> OP_3 OP_CHECKMULTISIG where metadata1 & metadata2 are pretend compressed public keys contain information about the contract
--------	---

- This sends the token back to the Merchant
- Output 2 - pays amount owed to Merchant
This amount should be monotonically increasing with each new Replaceable Tx
- Output 3 - pays remainder of deposit to Customer

Fig. 6B

Customer Refund Transaction Details

Customer Refund Transaction	
CUSTOMER_REFUND_TX	
1	
FUNDING_TX	
IDX_001	
OP_1 <Sig_Customer> <serialized script customer>	
<ScriptSignatureFlags>	
0x00000001	
1	
450,000	
OP_DUP OP_HASH160 <Address Customer> OP_EQUALVERIFY	
OP_CHECKSIG	
N	

Fig. 7A

- Input 1 - spends Output 1 of Funding Tx using Customer sig to unlock <serialized script customer>:

Script	OP_IF <now + time N> OP_CHECKLOCKTIMEVERIFY OP_DROP <PubKey_Customer> OP_CHECKSIGVERIFY OP_ELSE <PubKey_Customer> <PubKey_Merchant> OP_2 OP_CHECKMULTISIGVERIFY OP_ENDIF
--------	---

- Note that the Sequence Number is not 0xFFFFFFFF which signals for Locktime N to be checked
- Output 1 - pays the amount in Funding Tx Output 1 to whatever addresses Customer chooses to nominate

Fig. 7B

Merchant Refund Transaction Details

Merchant Refund Transaction	
MERCHANT_REFUND_TX	
1	
FUNDING_TX	
IDX_002	
OP_1 <Sig_Merchant> <serialized script merchant>	
<ScriptSignatureFlags>	
0x00000001	
1	
50,000	
OP_DUP OP_HASH160 <Address Customer> OP_EQUALVERIFY	
OP_CHECKSIG	
N	

Fig. 8A

- Input 1 - spends Output 1 of Funding Tx using Merchant sig to unlock <serialized script merchant>:

Script	OP_IF <now + time N> OP_CHECKLOCKTIMEVERIFY OP_DROP <PubKey_Merchant> OP_CHECKSIGVERIFY OP_ELSE <PubKey_Customer> <PubKey_Merchant> OP_2 OP_CHECKMULTISIGVERIFY OP_ENDIF
--------	---

- Note that the Sequence Number is not 0xFFFFFFFF which signals for Locktime N to be checked
- Output 1 - pays the amount in Funding Tx Output 2 to whatever addresses Merchant chooses to nominate
 - Merchant can choose to use one of the outputs to remint the token

Fig. 8B

Minting Transaction	
MINTING_TX	
1	
<Tx containing Funds by Hotel in address H, 100,000,000 Satoshis>	
IDX_00	
<Sig H> <PubKey H>	
<ScriptSignatureFlags>	
0xFFFFFFFF	
5	
99,994,000	
OP_DUP OP_HASH160 <Address H> OP_EQUALVERIFY	
OP_CHECKSIG	
1,000	
OP_HASH160 <script_hash_token1> OP_EQUAL	
1,000	
OP_HASH160 <script_hash_token2> OP_EQUAL	
1,000	
OP_HASH160 <script_hash_token3> OP_EQUAL	
1,000	
OP_HASH160 <script_hash_token4> OP_EQUAL	
0	

Fig. 9A

Output 1 - P2PKH. Pays the remaining funds back to Hotel's address H

Output 2 - P2SH (Token 1). Pays 1,000 Satoshi to the hash of the script
<script_hash_token3>:

Script	OP_1 <Metadata1> <Metadata2> <PubKey H> OP_3 OP_CHECKMULTISIG where metadata1 & metadata2 are pretend compressed public keys contain information about contract 1
--------	--

Fig. 9B

Funding Transaction
FUNDING_TX
2
<Tx containing Funds by Guest in address G, 6,000,000 Satoshis>
IDX_00
<Sig G> <PubKey G>
<ScriptSignatureFlags>
0xFFFFFFFF
MINTING_TX
IDX_004
<Placeholder for Sig H> <script_serialized_token3>
<ScriptSignatureFlags>
0xFFFFFFFF
2
3,000,000
OP_HASH160 <script_hash_guest> OP_EQUAL
3,000,000
OP_HASH160 <script_hash_hotel> OP_EQUAL
0

Fig. 10A

Input 2 - Copy of the token details on Hotel's intranet site.
Requires the Hotel's signature to be complete

Output 1 - P2SH. Pays 3,000,000 to the hash of the script
<script_hash_guest>:

Script	OP_IF <now + 48hrs> OP_CHECKLOCKTIMEVERIFY OP_DROP <PubKey G> OP_CHECKSIGVERIFY OP_ELSE <PubKey G> <PubKey H> OP_2 OP_CHECKMULTISIGVERIFY OP_ENDIF
--------	--

The locked funds can be claimed by the Guest in 48 hours time
if not spent.

Output 2 - P2SH. Pays 3,001,000 to the hash of the script
<script_hash_hotel>:

Script	OP_IF <now + 48hrs> OP_CHECKLOCKTIMEVERIFY OP_DROP <PubKey H> OP_CHECKSIGVERIFY OP_ELSE <PubKey G> <PubKey H> OP_2 OP_CHECKMULTISIGVERIFY OP_ENDIF
--------	--

The locked funds can be claimed by the Hotel in 48 hours time.
This is the minimum payment due as per Contract 3 and the
value of the token itself.

Fig. 10B

Replaceable Transaction	
REPLACEABLE_TX	
2	
FUNDING_TX	
IDX_001	
OP_0 <Sig G> <Placeholder for Sig H> <serialized script guest>	
<ScriptSignatureFlags>	
0xFFFFFFFF	
FUNDING_TX	
IDX_002	
OP_0 <Sig G> <Placeholder for Sig H> <serialized script hotel>	
<ScriptSignatureFlags>	
0xFFFFFFFF	
3	
1,000	
OP_HASH160 <script_hash_token3> OP_EQUAL	
3,025,000	
OP_DUP OP_HASH160 <Address H> OP_EQUALVERIFY	
OP_CHECKSIG	
2,975,000	
OP_DUP OP_HASH160 <Address C> OP_EQUALVERIFY	
OP_CHECKSIG	
0	

Fig. 11A

Input 1-2 - Spends the Funding Transaction outputs. Requires the Hotel's signature to be complete

Output 1 - P2SH. Sends the token back to Hotel. Pays 1,000 to the hash of script `<script_hash_token3>`

Output 2 - P2PKH. Payment owed to Hotel $3,000,000 + 25,000/\text{min} \times 1\text{min}$

Output 3 - P2PKH. Change back to Guest $6,000,000 - \text{payment to Hotel}$

Fig. 11B