

DApp marketplace

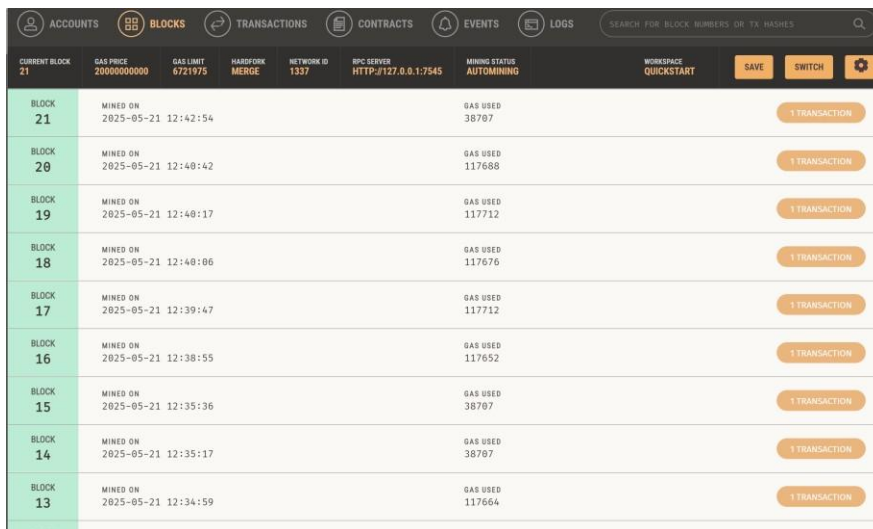
Report

Εργαλεία που χρησιμοποιήθηκαν: Ganache, MetaMask, Web3.js, Remix.

Επισκόπηση και σύντομη περιγραφή:

Αναπτύχθηκε μια web-DApp “Summer Marketplace” που τρέχει σε τοπικό δίκτυο Ganache. Ο χρήστης συνδέεται με MetaMask, βλέπει το ενεργό address στη σελίδα και:

- List Product – καταχωρεί νέο προϊόν (όνομα και τιμή σε ETH) η τιμή μετατρέπεται σε wei εσωτερικά και αποθηκεύεται στο smart contract.
- Πίνακας Listings – φέρνει από το contract όλα τα διαθέσιμα items του ίδιου chain ID και τα εμφανίζει σε πραγματικό χρόνο.
- Buy – κάθε γραμμή έχει κουμπί Buy όπου ο αγοραστής υπογράφει τη συναλλαγή, στέλνει ακριβώς το ποσό σε wei και το item μαρκάρεται ως **sold**.
- Όλα τα δεδομένα (προϊόντα, πωλήσεις) μένουν στην αλυσίδα – οποιοσδήποτε λογαριασμός στο ίδιο δίκτυο βλέπει το ίδιο status. Επιβεβαίωση αυτού μέσω Ganache που φαίνονται όλα τα transactions/ Blocks.



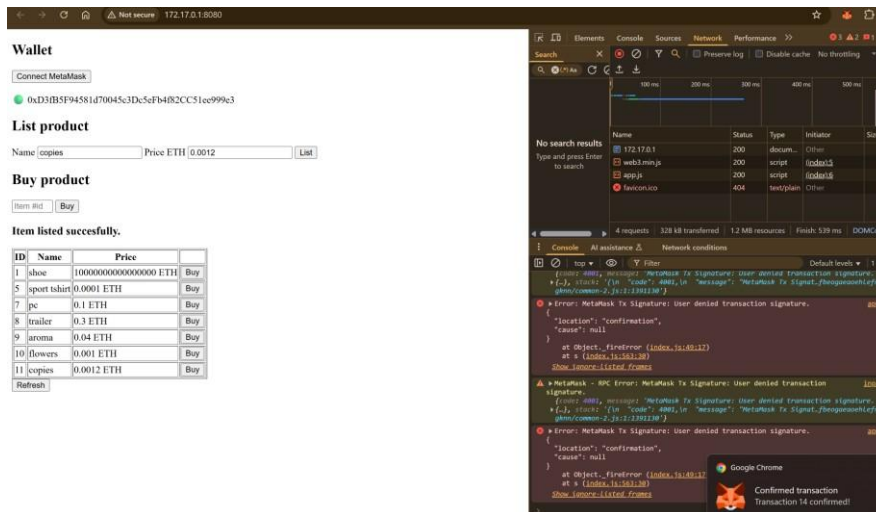
CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER	MINING STATUS	WORKSPACE
21	2000000000	6721975	MERGE	1337	HTTP://127.0.0.1:7545	AUTOMINING	QUICKSTART

BLOCK	MINED ON	GAS USED	TRANSACTION
21	2025-05-21 12:42:54	38707	1 TRANSACTION
20	2025-05-21 12:40:42	117688	1 TRANSACTION
19	2025-05-21 12:40:17	117712	1 TRANSACTION
18	2025-05-21 12:40:06	117676	1 TRANSACTION
17	2025-05-21 12:39:47	117712	1 TRANSACTION
16	2025-05-21 12:38:55	117652	1 TRANSACTION
15	2025-05-21 12:35:36	38707	1 TRANSACTION
14	2025-05-21 12:35:17	38707	1 TRANSACTION
13	2025-05-21 12:34:59	117664	1 TRANSACTION

Το front-end είναι απλό HTML/JS με Web3.js, το back-end ένα Solidity smart-contract (SummerMarketplace.sol) που διαχειρίζεται τα listings και κάνει κάποιους ελέγχους όπως require για τιμή > 0, έλεγχος sold.

Η list product φορμα πραγματοποιεί καταχώρηση νέων items (όνομα item, τιμή σε ETH

(αποθηκεύεται σε wei)), η buy product – πραγματοποιεί αγορά βάζοντας το Id του προϊόντος (item) που ενδιαφέρει, και ο πίνακας που φέρνει όλα τα items που έχουν καταχωρηθεί για το συγκεκριμένο Chain ID /contract address.



Ganache accounts (αντιγράφουμε το private key για testing του Dapp από όσα accounts θέλουμε minimum πχ. 2 για να προσομοιώσουμε αγοραπωλησία) και συνδεόμαστε στο metamask ανάλογα κάθε φορά. Ένα γενικό σχήμα θα ήταν να δημιουργήσουμε listing με account1 και αγορά με το account2. Βλέπουμε παρακάτω ανάλογα με τα transactions που έχουν πραγματοποιηθεί, να έχουν διαμορφωθεί και τα balances.

ACCOUNTS
 BLOCKS
 TRANSACTIONS
 CONTRACTS
 EVENTS
 LOGS

CURRENT BLOCK
21

GAS PRICE
20000000000

GAS LIMIT
6721975

HARDFORK
MERGE

NETWORK ID
1337

RPC SERVER
HTTP://127.0.0.1:7545

MINING STATUS
AUTOMINING

WORKSPACE
QUICKSTART

SAVE

SWITCH

MNEMONIC

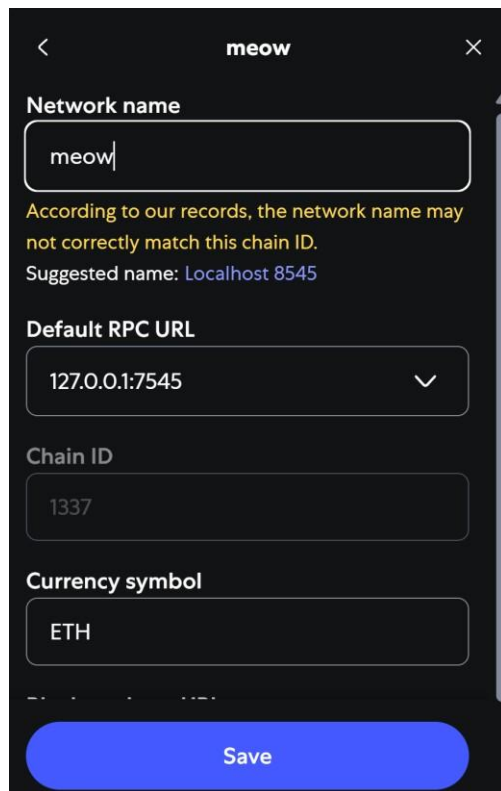
HD PATH

aunt distance deal shoulder milk patrol flash enter talent melt either volume

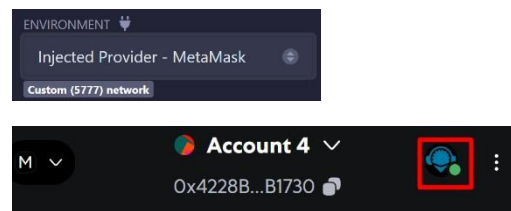
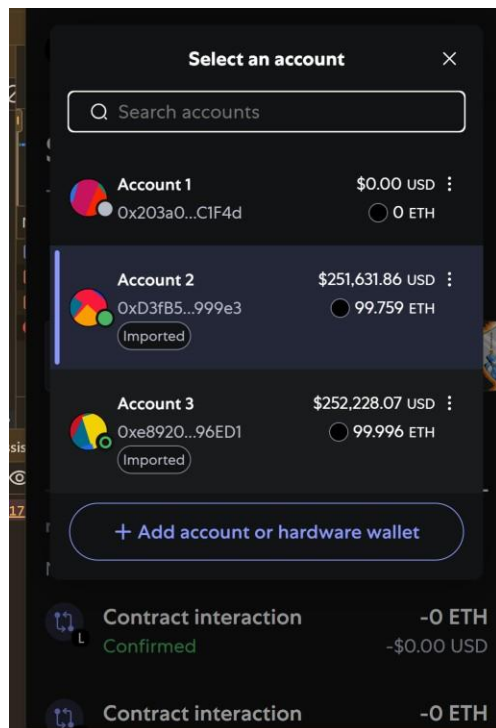
m44'60'o'0account_index

ADDRESS	BALANCE	TX COUNT	INDEX	
0xD3fB5F94581d70045c3Dc5eFb4f82CC51ee999e3	99.76 ETH	15	0	
0xe89206CbC4ce47e7BEDC981529Cea8A137296ED1	99.99 ETH	6	1	
0xC22e7DBf1D1c9ED75374FfeBAC8bf9645f034D24	100.00 ETH	0	2	

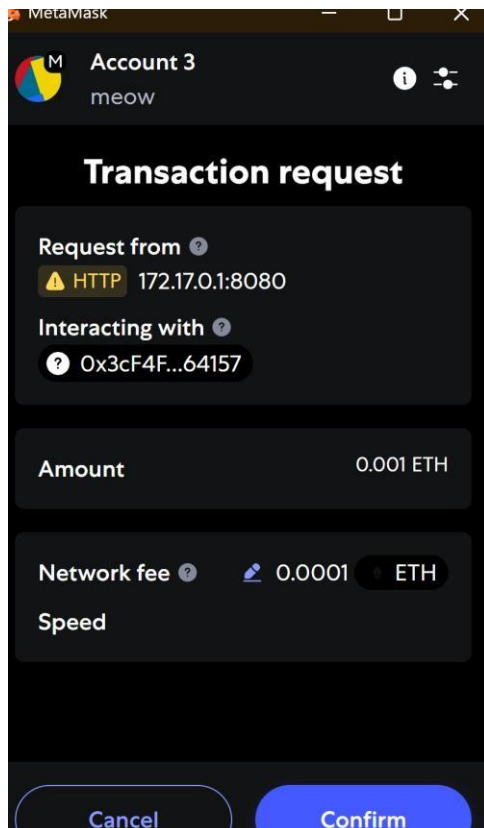
Προηγούνται τα βήματα της διασύνδεσης του Remix με το metamask για το chain id που ενδιαφερομαστε και το RPC URL = χρήση του test network που φτιάξαμε.



Επιλογή / Δημιουργία account



Πραγματοποιούμε ένα οποιοδήποτε transaction – interaction with Metamask – παράδειγμα αγορά :



Αποτέλεσμα στον browser

Wallet

Connect MetaMask

0xe89206CbC4ce47c7BEDC981529Cea8A137296ED1

List product

Name Price ETH

Buy product

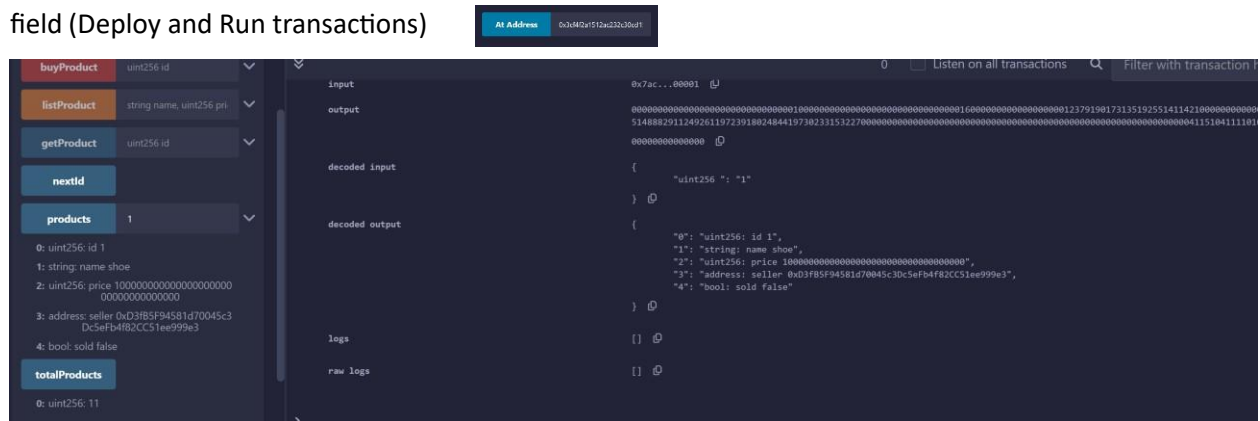
Item #id

✓ bought item : 10

ID	Name	Price	
1	shoe	10000000000000000000 ETH	<input type="button" value="Buy"/>
5	sport tshirt	0.0001 ETH	<input type="button" value="Buy"/>
7	pc	0.1 ETH	<input type="button" value="Buy"/>
8	trailer	0.3 ETH	<input type="button" value="Buy"/>
9	aroma	0.04 ETH	<input type="button" value="Buy"/>
11	copies	0.0012 ETH	<input type="button" value="Buy"/>
<input type="button" value="Refresh"/>			

Μπορούμε να δούμε τα products από το remix επίσης, γράφοντας το contract address στο At Address

field (Deploy and Run transactions)



Δομή του smart contract SummerMarketplace

```
struct Product { id, name, price(wei), seller, sold } και mapping(uint  $\Rightarrow$  Product) products
```

Καταχώρηση:

`listProduct(name, priceWei)`: διαβάζει input σε ETH, το μετατρέπει σε wei (**toWei**) και καλεί **contract.methods.listProduct** (contract's function).

`require(price > 0)` → δημιουργεί προϊόν, αποθηκεύει σε `wel`, δίνει διαδοχικό `id`.

Αγορά : `buyProduct(id).send({from:user, value: prod.price})`

buyProduct(id) payable:

Ελέγχει ότι το item δεν έχει πουληθεί και ότι msg.value == product.price. Θέτει sold=true και μεταφέρει τα ETH στον seller όταν η συναλλαγή είναι επιτυχής. from:user = η διεύθυνση που πληρώνει. value: prod.price = στέλνει ακριβώς τόσα wei όσα κοστίζει το προϊόν.

Πορεία: Ο πίνακας φιλτράρει το id και το priceWei στο buyProduct, MetaMask υπογράφει, Ganache εξορύσσει, το UI ανανεώνεται (statuz).

buyRow: κουμπί αγοράς στον πίνακα με τα listings. Τα listings που εμφανίζονται όλα υπάρχουν και αν όχι το transaction θα αποτύχει. Έχουμε ήδη το priceWei. Εκτελεί την buyProducts του contract.

Βοηθητικά functions:

```
getProduct(id)
```

totalProducts(): για χρήση από το js/ html αρχείο. Στο loadListings() εκτελεί totalProducts() και getProduct(id) για κάθε id.

Web3.js

```
async function connectWallet() {  
  if (!window.ethereum) return alert("Install MetaMask");  
  
  web3 = new Web3(window.ethereum);  
  await window.ethereum.request({ method: "eth_requestAccounts" });  
  user = (await web3.eth.getAccounts())[0];  
  account.innerText = `● ${user}`;  
  contract = new web3.eth.Contract(contractABI, contractAddress);  
}
```

Challenges:

1. Internal JSON-RPC error κατά την ανάπτυξη, δεν ήταν bug του συμβολαίου αλλά εμφανιζόταν όποτε οι ρυθμίσεις MetaMask, Ganache, Remix δεν ταίριαζαν (λάθος RPC URL/chain-ID, πολύ gas, λάθος contract address , ABI στο js).
2. Gas estimate timeout: οφειλόταν στο ότι ο λογαρισμός που προσπαθούσα να κάνω deploy δεν είχε ETH, άρα η eth_estimateGas απέτυχε.
3. Χρειάζεται redeploy στην επανεκκίνηση του Ganache αν χρειάζεται καινούργια έκδοση συμβολαίου.
4. Κάποιες φορές το MetaMask κολλούσε και δεν έκανε fetch το balance από accounts.
5. Έπρεπε να αλλάξει το ENV Version από advanced configuration στον compiler Remix σε κάτι άλλο πχ, paris, shanghai πέρα από το default Prague γιατί δεν υποστηριζόταν από Ganache.