

Marmottoshis

777 NFTs – 21 modèles en 37 exemplaires, tous backés par du BTC accumulé chaque mois en DCA.

[Voir le site Internet](#) 🌐

Liens directs

DApp : <https://www.marmottoshis.app>



Imaginé par **DocMarmott**, ce projet atypique, fruit d'une **collaboration avec pas moins de 21 artistes** francophones, permet d'obtenir des **NFTs dont la valeur augmente au fil du temps** grâce à un **DCA** mensuel et à un **supply** qui ne peut que se réduire.

L'idée étant d'injecter les fonds récoltés lors du mint initial en **DeFi** ce qui permet petit à petit (à chaque DCA) d'augmenter la valeur intrinsèque des NFTs. L'originalité ne s'arrête pas là, pour récupérer les **Satoshis adossés** à un NFT, l'utilisateur doit le « **burn** » (brûler), diminuant par conséquent le **supply global** (d'une part) mais aussi le **supply de son propre ID** (les tokens étant de type **ERC-1155**, un token dispose de plusieurs exemplaires). **Plus le temps passe et plus les utilisateurs restants profitent d'un DCA amélioré !**

Je ne peux que recommander RpGmAx comme dev web3, au-delà de comprendre votre projet et vos besoins il sait être force de proposition pour augmenter la proposition de valeur de vos objectifs.

Passionné de technique il ne se repose pas sur ses acquis (qui sont déjà immenses) et explore sans cesse de nouvelles fonctionnalités/techniques afin de toujours vous proposer de meilleurs services en adéquations avec vos attentes.

Disponible et réactif il vous accompagne dans les différentes phases de votre projet tout en assurant un suivi de l'existant. Il est sans aucun doute l'un des piliers de la réussite du projet Marmottoshis dont je suis le fondateur.

DocMarmott (fondateur)

Réflexions techniques

Ce projet est un **véritable concentré de technologies diverses**, l'expertise de chacun a été mise à contribution afin de permettre d'obtenir quelque chose d'unique et d'exploitable !

Stockage de métadonnées on-chain (permettant un reveal progressif), adossement/répartition des Satoshis, Dapp évolutive, nombreux pièges à éviter et stress tests intenses du **Smart Contract** et de la **DApp**... Les brainstormings furent nombreux et toujours très productifs et ce via une méthode Agile implicite 😊

Supply : 777/777



Edmond

Marketplace

37 exemplaires en circulation
35548 Satoshis adossés



Hector

Marketplace

37 exemplaires en circulation
35548 Satoshis adossés



Mariotte

Marketplace

37 exemplaires en circulation
35548 Satoshis adossés



Georges

Marketplace



Thodias

Marketplace



Teresa

Marketplace

Réalisation de la DApp évolutive

Plus poussée que la plupart des DApps, cette dernière regroupe de nombreuses fonctionnalités.

Qu'il s'agisse du **mint**, qui fut proposé via différentes étapes (free mint, réservations on-chain via page dédiée, listes blanches...) mais aussi du **dashboard** permettant d'obtenir en un clin d'œil l'intégralité des informations **on-chain** !

La Dapp offre aussi la possibilité de **burn** son NFT en spécifiant une adresse **Bitcoin** qui recevra les précieux **Satoshis**.

Cette dernière n'est pas figée dans le marbre et se veut évolutive, ce fut le cas entre les différentes phases et ce sera probablement encore le cas par la suite (il y a toujours de nouvelles idées à déployer) !

```
Extrait de la DApp Marmottoshis

const burntoBTCAddress = async (nft_family_id, bitcoinAddress, sats) => {
  if (!signerData) return;
  try {
    const nftContract = new ethers.Contract(contractAddress, abi, signerData);
    let nftTxn;

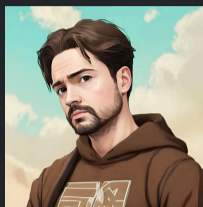
    const toast_loading = toast.loading(() => (<div><small>Burn d'un exemplaire du <b>#{nft_family_id}</b><br />avec redeem de <b>{sats} sats</b> à destination de l'adresse BTC :<br /><b>{bitcoinAddress}</b></small></div>), {style: { wordBreak: 'break-all' }});

    nftTxn = await nftContract.burnAndRedeem(nft_family_id, bitcoinAddress);

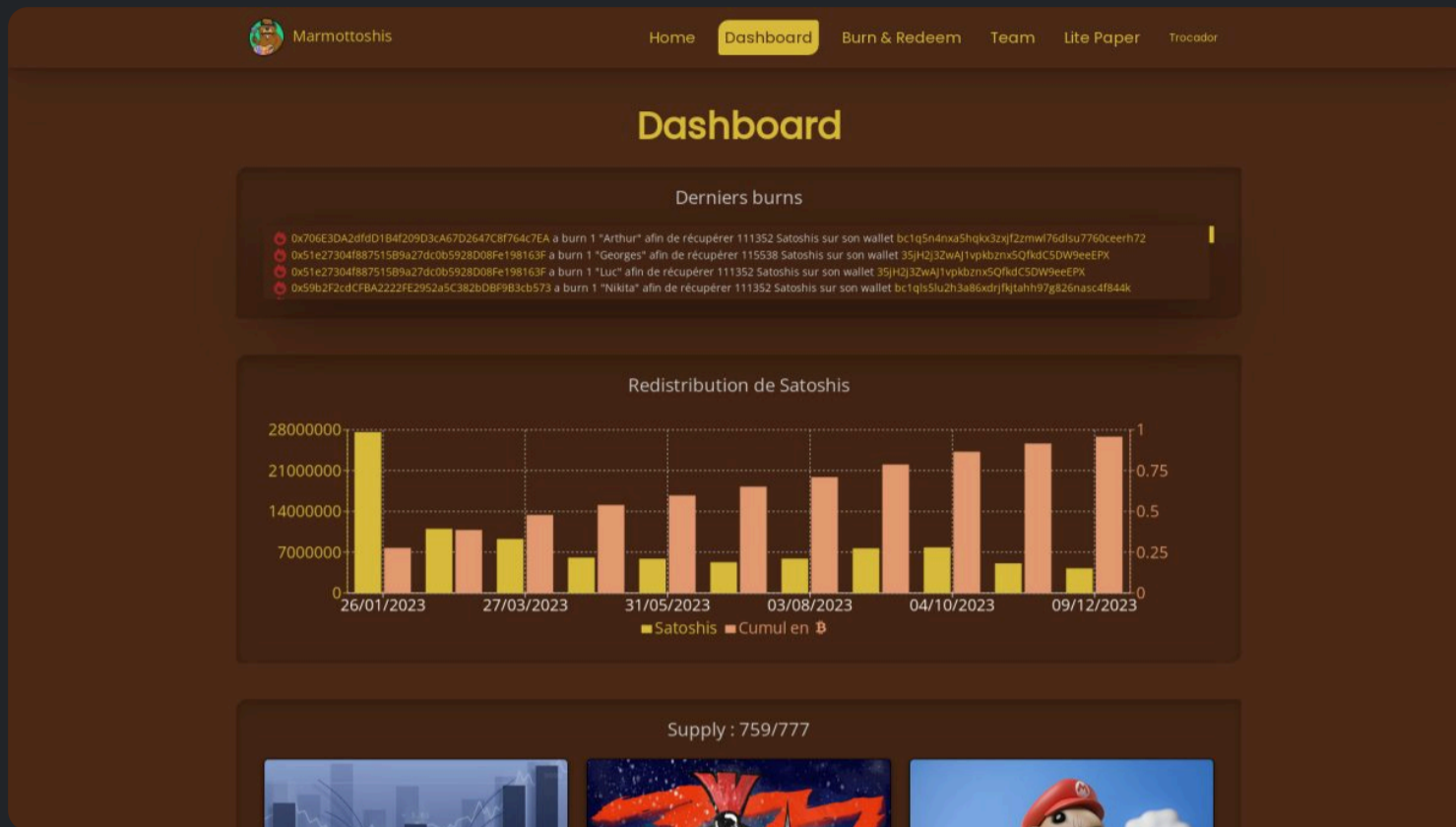
    toast.loading('Burn du NFT #' + nft_family_id + ' en cours', {id: toast_loading});
    await nftTxn.wait();

    toast.dismiss(toast_loading);
    toast.success('Burn du NFT #' + nft_family_id + ' réussi !');

    getUserNFTs();
  } catch (err) {
    toast.dismiss();
    toast.error('Burn du NFT #' + nft_family_id + ' annulé !');
    console.log(err);
    if (err.code === 'INSUFFICIENT_FUNDS') { toast.error('Fonds insuffisants'); }
  }
}
```



© 2025. Site sans cookie intrusif (aucun suivi des visites)



Marmottoshis v2

Un projet évolutif, à la manière de la rareté de ses NFTs.

[Voir le site Internet](#) ☺

Liens directs

DApp : <https://www.marmottoshis.app>



Prolongement direct de la **v1** publiée début 2023, cette version 2 apporte son lot de nouveautés !

- Bascule d'Ethers.js vers **viem** afin d'assurer une totale **pérennité** à la DApp (connecteur de wallet, fonctionnalités...).
- Dashboard évolué avec **suivi dynamique** de la redistribution de satoshis et cumul en BTC.
- **Optimisation** conséquente du code du dashboard afin d'obtenir des statistiques de manière instantanée.
- Dev d'un serveur **Node.js** de parsing des **données** on-chain.

- Dev d'un serveur **Node.js** dédié à la **notification** sur Discord en cas de burn.

Spoiler : Une version 3 apportant une extension au projet est prévue.

Création d'un serveur de stats Node.js

Afin de palier à l'absence de certaines informations en provenance du Smart Contract (car non prévues initialement), un serveur Node.js a été réalisé.

Ce dernier permet de parser les différentes transactions initiées par un wallet spécifique et à destination du Smart Contract. Il retourne ainsi le nombre de **satoshis redistribués** & le **cumul en BTC** ainsi que les derniers **burns**.

La DApp est par conséquent rattachée à celui-ci et affiche des informations continuellement à jour.

```
Extrait du serveur Node.js de stats

app.get('/satoshis', cacheMiddleware(300), async (req, res) => {

  // On récupère l'ensemble des tx à destination du SC et initiées par le wallet qui nous intéresse (Doc)
  try {
    const data = await alchemy.core.getAssetTransfers({
      fromBlock: "0x0",
      contractAddresses: [contractAddress],
      fromAddress: docAddress,
      excludeZeroValue: false,
      withMetadata: true,
      category: ["external"],
    });

    let transactions = (await Promise.all(data.transfers.map(getTransactionInfos))).filter(Boolean);

    let cumul = 0;
    for (let i = 0; i < transactions.length; i++) {
      cumul += parseInt(transactions[i].Satoshis);
      transactions[i]['Cumul en B'] = sb.toBitcoin(cumul);
    }

    console.log(transactions);

    res.json({ transactions });

  } catch (err) {
    console.error(err);
    res.status(500).json({
      message: 'An error occurred.'
    });
  }
});
```

Et de son "petit frère" dédié aux notifications Discord

Afin d'**alerter automatiquement** l'ensemble des membres du projet, une notification **Discord** est émise dès qu'un burn est effectué. Cette dernière affiche le **nom** du Marmottoshi « brûlé », la **transaction** associée et les **satoshis** récupérés par l'auteur du burn.

Une manière simple et efficace de relayer de l'information on-chain.

🔥 Nouveau burn

Marmottoshi brûlé :

Arthur

Transaction :

<https://etherscan.io/tx/0xf3916ea7eabead0c67aa398e1cbff22cc94776ef727aa681f6c31e984462fb0f>

Satoshis récupérés :

111352 Sats

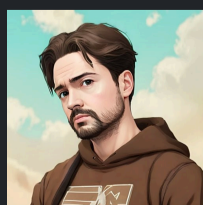


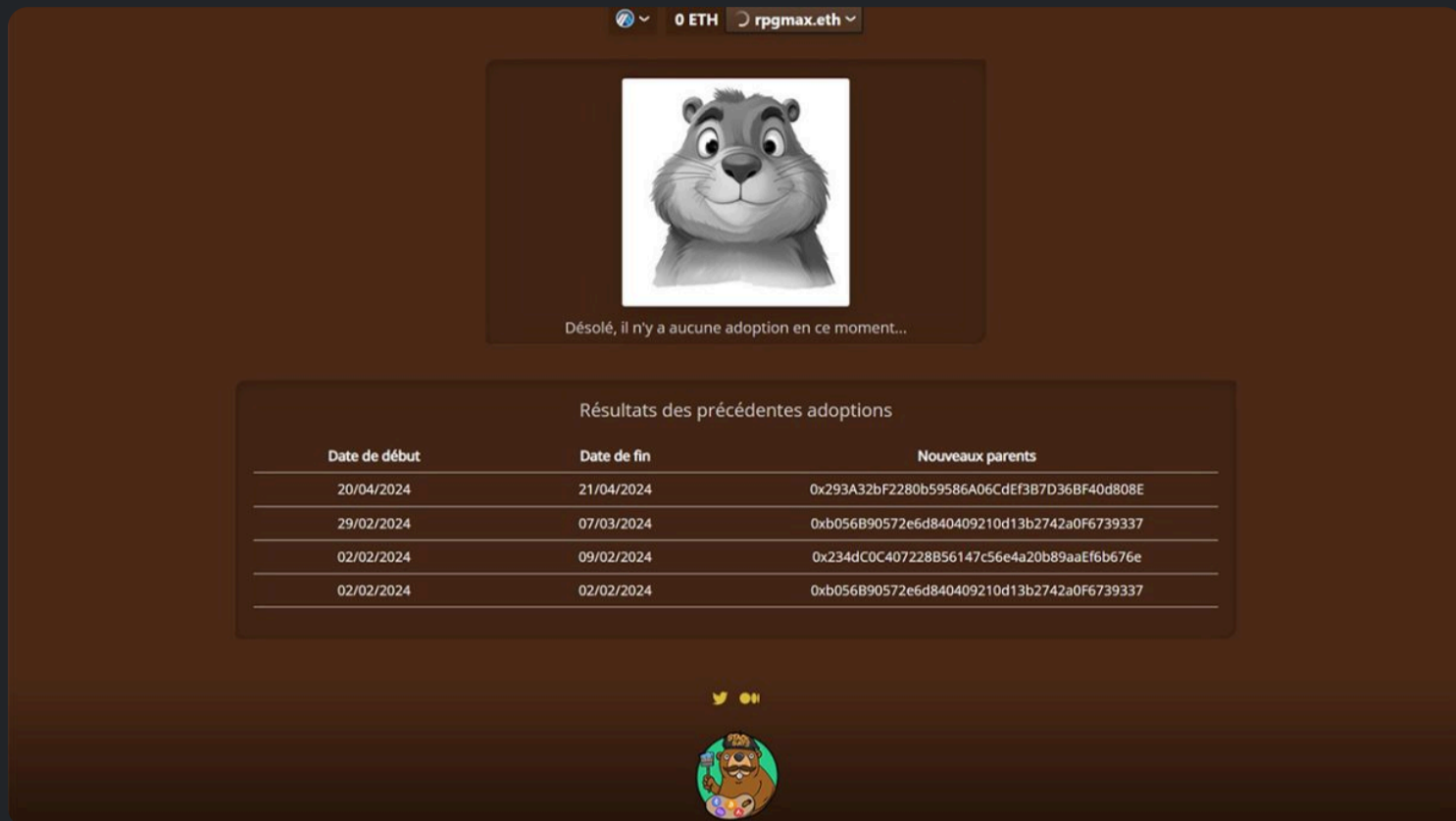
Le marmottobot retourne dans son terrier... 🌿



Extrait du serveur Node.js de notification Discord

```
const embedsData = {
  embeds: [
    {
      title: '🔥 Nouveau burn',
      type: 'rich',
      url: 'https://www.marmottoshis.app/dashboard',
      footer: {
        text: 'Le marmottobot retourne dans son terrier... 🌿',
      },
      image: {
        url: 'https://www.marmottoshis.app/images/discord/alert.jpg',
      },
      fields: [
        {
          name: 'Marmottoshi brûlé :',
          value: data.name,
          inline: true,
        },
        {
          name: 'Transaction :',
          value: `https://etherscan.io/tx/${data.tx}`,
          inline: false,
        },
        {
          name: 'Satoshis récupérés :',
          value: `${data.satoshis} Sats`,
          inline: true,
        },
      ],
    },
  ],
};
```





Marmottoshis v4

Une évolution sans fin 😊

Liens directs

DApp : <https://www.marmottoshis.app>

Smart Contract : Adopt A Marmot



On saute une version en passant directement de la **v2** à la **version 4** du projet !

En effet, pas mal de choses depuis, dont l'implémentation d'une fonctionnalité bientôt disponible qui justifie le passage en v4 😊

- **Implémentation d'une nouvelle fonctionnalité « Adopt A Marmot »** avec son propre Smart Contract sur **Arbitrum**.
- Implémentation d'une future fonctionnalité « **MarmottoShow** » bientôt disponible...
- **Migration importante des bibliothèques** utilisées vers une version majeure plus récente. Breaking change oblige, une bonne partie du code de la dApp a du être réécrit, à nouveau ><

- Réécriture puis fusion des serveurs **Node.js** implémentés dans la v2 avec pour objectif de décentraliser totalement ces derniers.
- Nombreuses optimisations du code et de l'UI. Comme toujours 😊

Toujours un réel plaisir de travailler avec RpGmAx, en veille technologique permanente, il est proactif sur l'existant tout en y intégrant des évolutions.

Doté de nombreuses compétences techniques il apporte également de la fluidité dans les échanges entre membre du staff technique.

Bref je ne peux que recommander les services de RpGmAx (tant que vous le laissez dispo quand j'en ai besoin :p)

DocMarmott (fondateur)

Adopt A Marmot

Nouvelle fonctionnalité déployée sur le réseau **Arbitrum**, **Adopt A Marmot** permet à la team du projet de mettre en « **adoption** » des marmottes (en stock ou rachetées sur le marché secondaire). Les **futurs parents** peuvent tenter de faire une adoption en effectuant un ou plusieurs dons (en fonction des conditions de l'adoption déterminées dans le **smart contract réalisé par mes soins**).

La team peut ensuite réaliser un tirage au sort qui va déterminer un heureux élu !

Mélangeant aléatoire et gamification, notamment au travers d'une **interface dynamique et visuelle** (avec une forte inspiration « Tamagotchi », désolé ^^), cette nouvelle fonctionnalité permet de créer un cercle vertueux entre vendeurs et acquéreurs à petit budget.

```
Marmottoshis - Adop A Marmot teasing

// [External] Donation function with many checks
function adoptRequestDonation(uint256 amount) external payable {
    require(_nextAdoptionID > 1, "No adoption available");
    require(isAdoptionPaused == false, "Adoption system is paused");

    uint256 adoptionId = getCurrentAdoptionId();
    Adoption storage currentAdoption = _adoptions[adoptionId];
    require(currentAdoption.endDate == 0, "Sorry, adoption ended...");
    require(amount > 0, "Mmmmh, seriously?");
    require(msg.value == (donationAmount * amount), "Incorrect ETH amount");

    require(
        _adoptionDonations[adoptionId][msg.sender] + amount <=
            maxDonationsPerWallet,
        "Limit reached"
    );

    _adoptionDonations[adoptionId][msg.sender] += amount; // For tracking purpose
    currentAdoption.donations += amount; // For informations only

    for (uint256 i = 0; i < amount; i++) {
        _adoptionDonors[adoptionId].push(msg.sender); // For the drawn
    }

    // LT tracking
    if (_allTimeDonations[msg.sender] == 0) {
        _allTimeDonors.push(msg.sender);
    }
    _allTimeDonations[msg.sender] += amount;

    emit DonationMade(msg.sender, msg.value);
}
```



Adoption en cours (débutée le 20/04/2024)

0.004 ETH par donation ~ 5 donations / wallet ~ 1 parent recherché

Actuellement 83 donations

Vous avez réalisé 0 / 5 donations possibles



Donations - 4 +

 Faire un don (0.016 ETH)

Il semblerait que vous soyez à court d'ETH dans votre wallet... N'hésitez pas à consulter la page Swap pour bridger vers Arbitrum.

MarmottoShow

Pas encore dévoilée publiquement, cette nouvelle fonctionnalité arrivera prochainement sur une troisième blockchain, faisant de **Marmottoshis** le projet réalisé avec le plus de blockchains cohabitant dans la même dApp !

J'ai eu le plaisir de créer une interface sobre et efficace adaptée, qui sera révélée le moment venu 😊



© 2025. Site sans cookie intrusif (aucun suivi des visites)