Signed Transfer Manager

- **Introduced in:** 2.1.0
- **Contract name: SignedTransferManager.sol**
- **Type:** Experimental - Transfer Manager Module

How it works

This module allows anyone making a `mintWithData` or `transferWithData` call to include signed data with the call that dynamically authorises the mint / transfer without the need for a separate on-chain call to a whitelist or manual approval.

Key functionalities (as defined in the Smart Contract)

Initialization

This module has no initialisation.

Signers

The module keeps a list of addresses that have been authorised to sign messages to dynamically authorise transfers.
The function:
`function updateSigners(address[] _signers, bool[] _signersStats) public withPerm(ADMIN)`
can be used to update the list of signers for this module.

Signing Transfers

In order to use the module, signed data must be submitted with the `mintWithData` or `transferWithData` functions.

The data must be signed by an authorised signer.

The signed data is:
[moduleAddress, fromAddress, toAddress, amount]

For an example of how to do this using web3, see:
`test/y_signed_transfer_manager.js`

Invalidating Signatures

A signature can only be used once (i.e. a user can't reuse the same signed data for multiple transfers).
If a signer wishes to invalidate their signature, the signer can use the function:
`function invalidSignature(address _from, address _to, uint256 _amount, bytes _data) public`
to do so.

Special considerations / notes

Note that the signed data has to include both the sender, receiver & token amount.
A valid signature will fully authorise the transfer unless the transfer is marked as INVALID by another module.

Troubleshooting / FAQs

None

Know Issues / bugs

Experimental module - intended as a proof of concept to exercise minting and transfers with data.

A signature can only be used once, meaning if the same transfer needs to be authorised multiple times different amount values must be used which makes this impractical for production use.

A more sophisticated use of `_data` would be to specify a schema involving the target module and nonce at the start of the `_data`.