

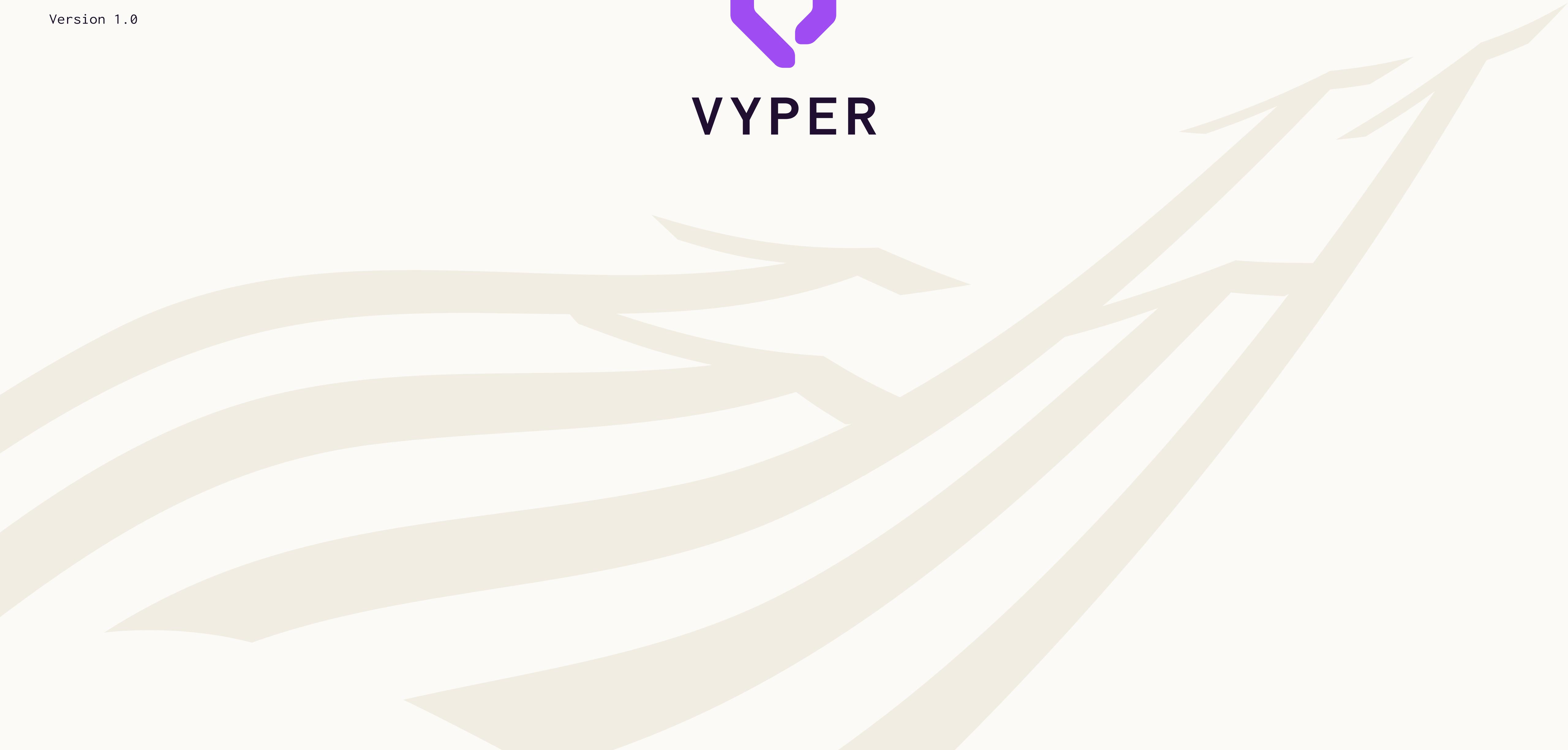
**VYPER**  
Visual Identity

# Toolkit

Version 1.0



**VYPER**



## Overview

### Logo



VYPER



### Color

Violet 9F4CF2	Violet 75% B779F5	Violet 50% CFA5F8	Violet 15% F1E4FD
Sand DBCBAB	Sand 75% E4D8C0	Sand 50% EDE5D5	Sand 15% FAF7F2
A horizontal color palette bar at the bottom of the color chart, showing a gradient from yellow-green to orange.			

### Typography

SemiExpanded Bold  
SemiExpanded Medium  
SemiExpanded Regular

AaBbCc  
0123

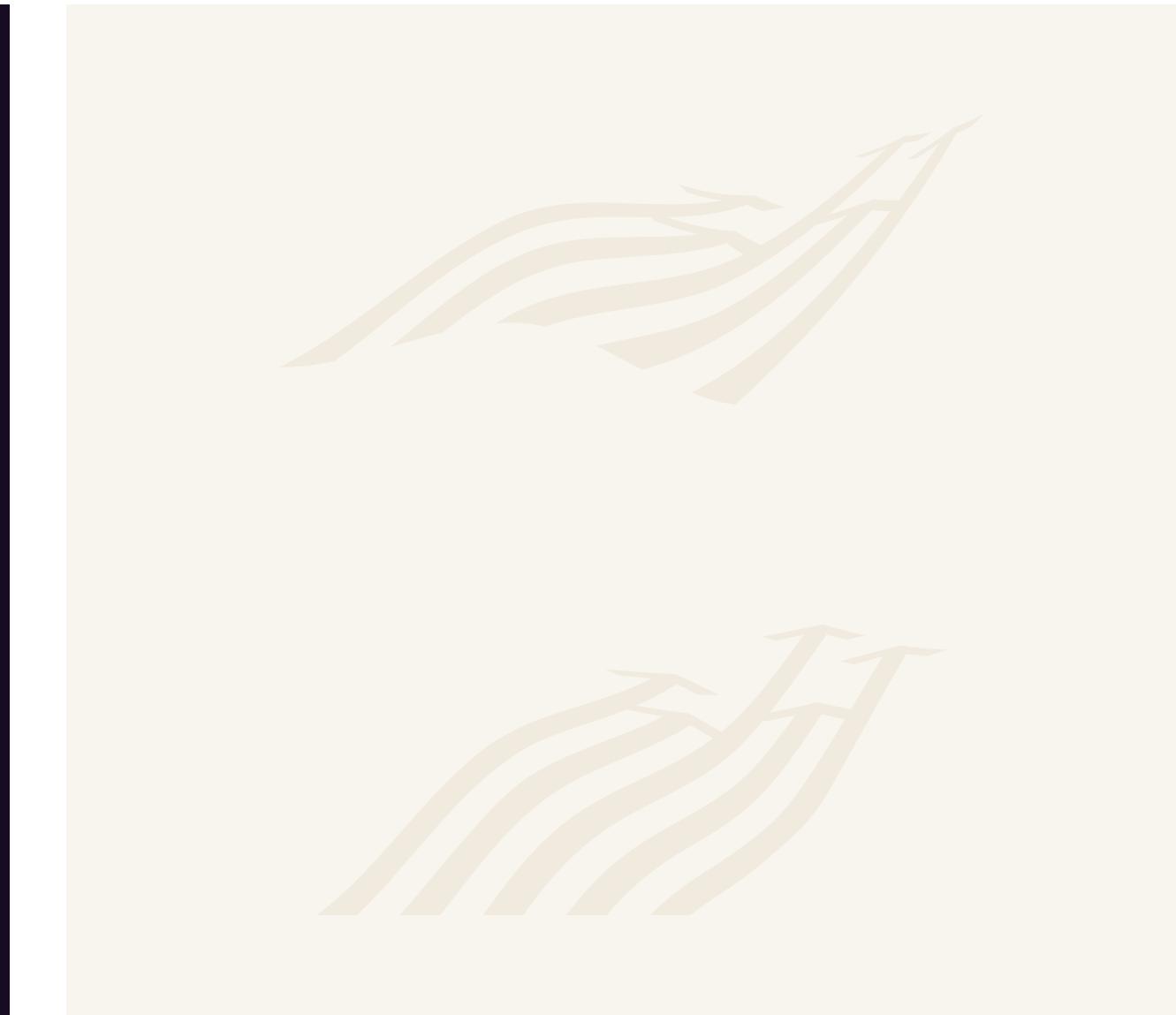
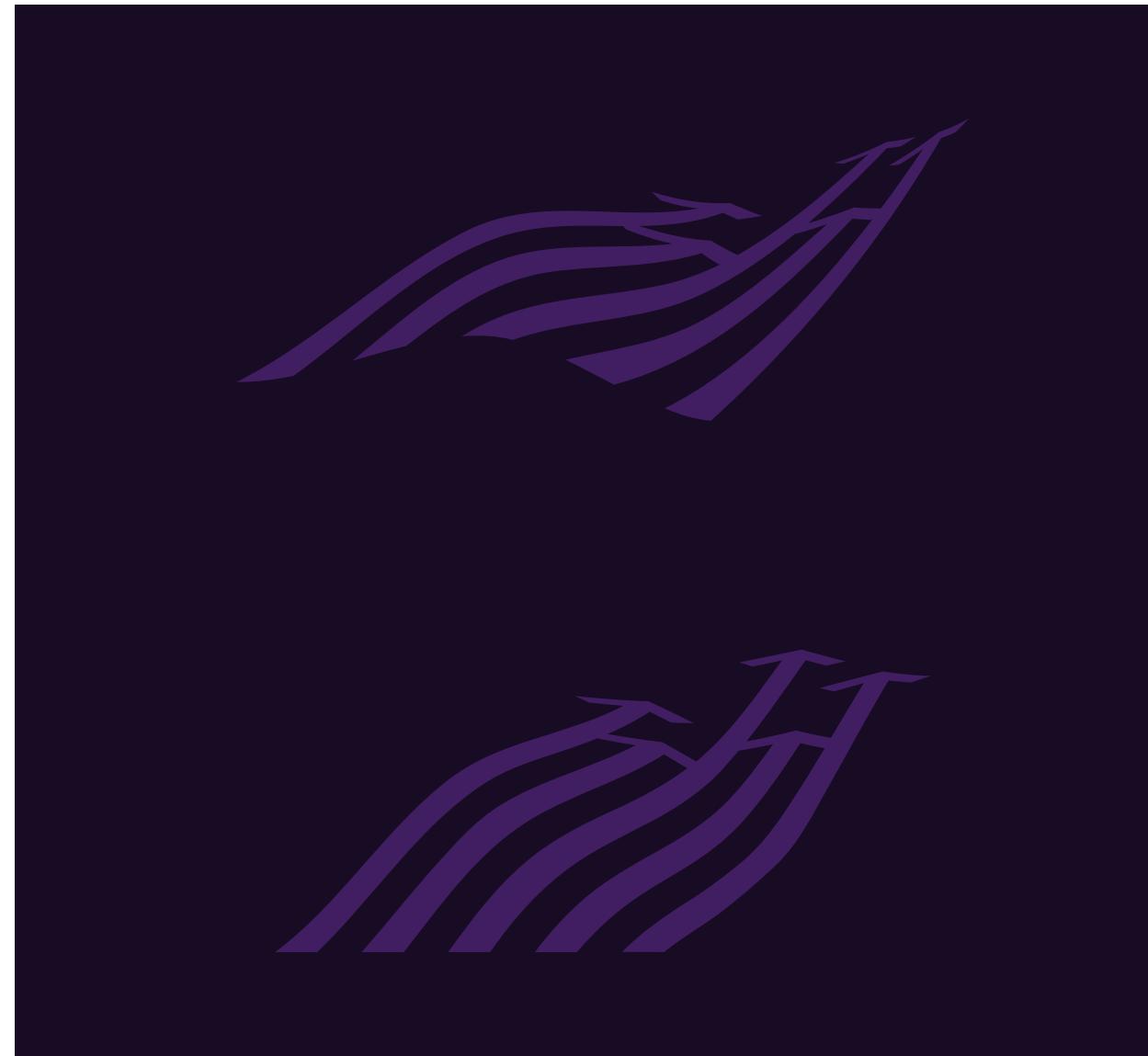
Inconsolata

Headlines set in  
SemiExpanded Bold

Highlighted paragraphs set in SemiExpanded Medium

Body copy set in SemiExpanded Regular.

### Supergraphic



### Applications



## Logo

### Color Portrait Logo

Our primary logo  
should be used  
wherever possible.

### Mono Portrait Logo

In cases where the  
logo can not be shown  
in color, use the  
Mono versions as  
shown here.

### Color portrait versions



VYPER

Positive



Negative

### Mono portrait versions



VYPER

Positive



Negative

## Logo

### Color Landscape Logo

Our secondary logo  
should only be used  
when there is  
insufficient space to  
use the primary logo.

### Mono Landscape Logo

In cases where the  
logo can not be shown  
in color, use the  
Mono versions as  
shown here.

### Color landscape versions



Positive



Negative

### Mono landscape versions



Positive



Negative

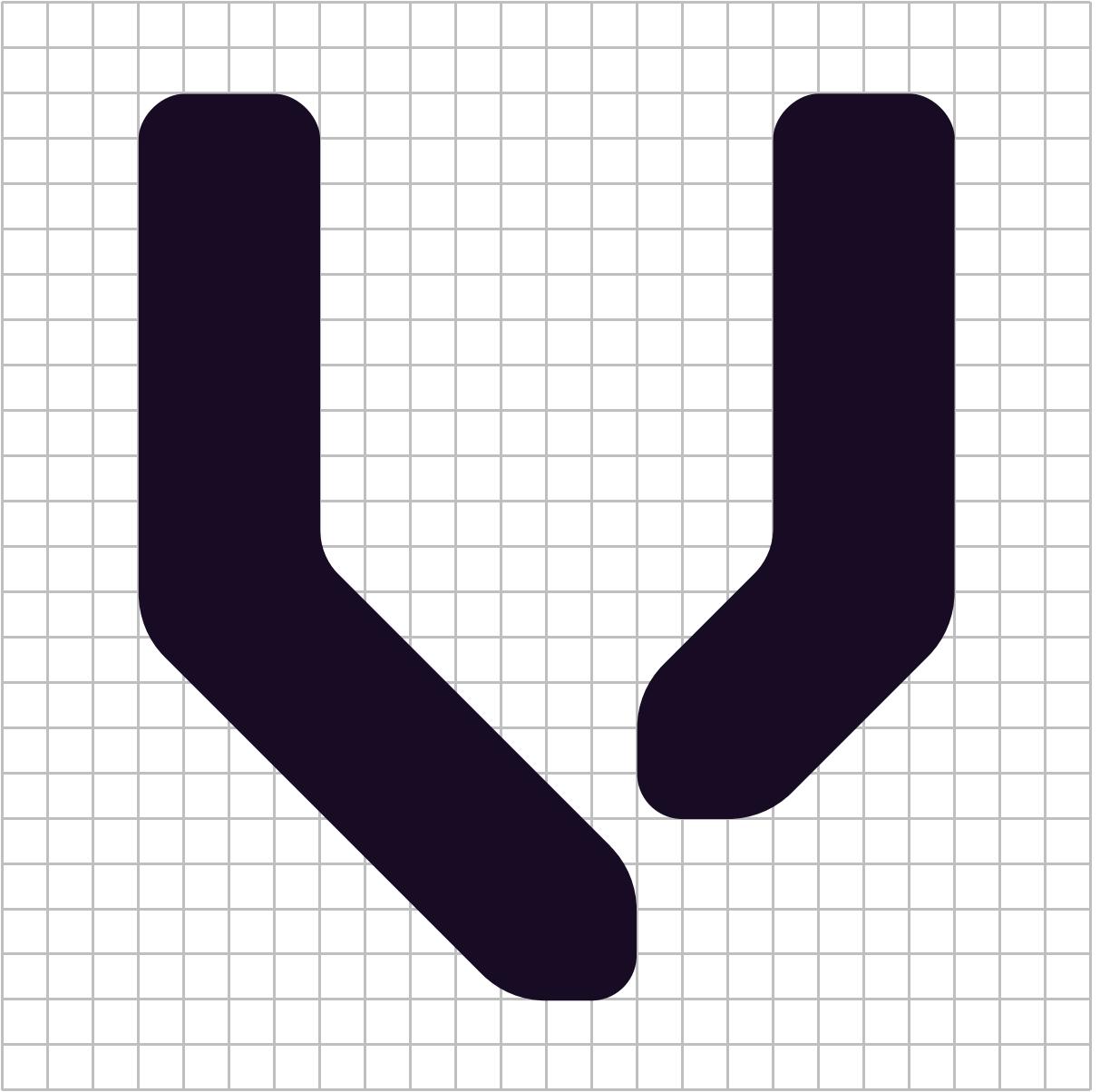
**Avatar****Symbol**

For symbols used on social channels, use the Violet and light Sand version shown here.

**Structure**

Live area 18w x 20h

Padding 3x, 2y



## Typography

Inconsolata is our typeface

Background info on  
Inconsolata can be found  
[here](#).

Download from [google](#).

**Inconsolata**  
AaBbCc  
0123

**SemiExpanded Bold**  
**SemiExpanded Medium**  
**SemiExpanded Regular**  
**Regular**

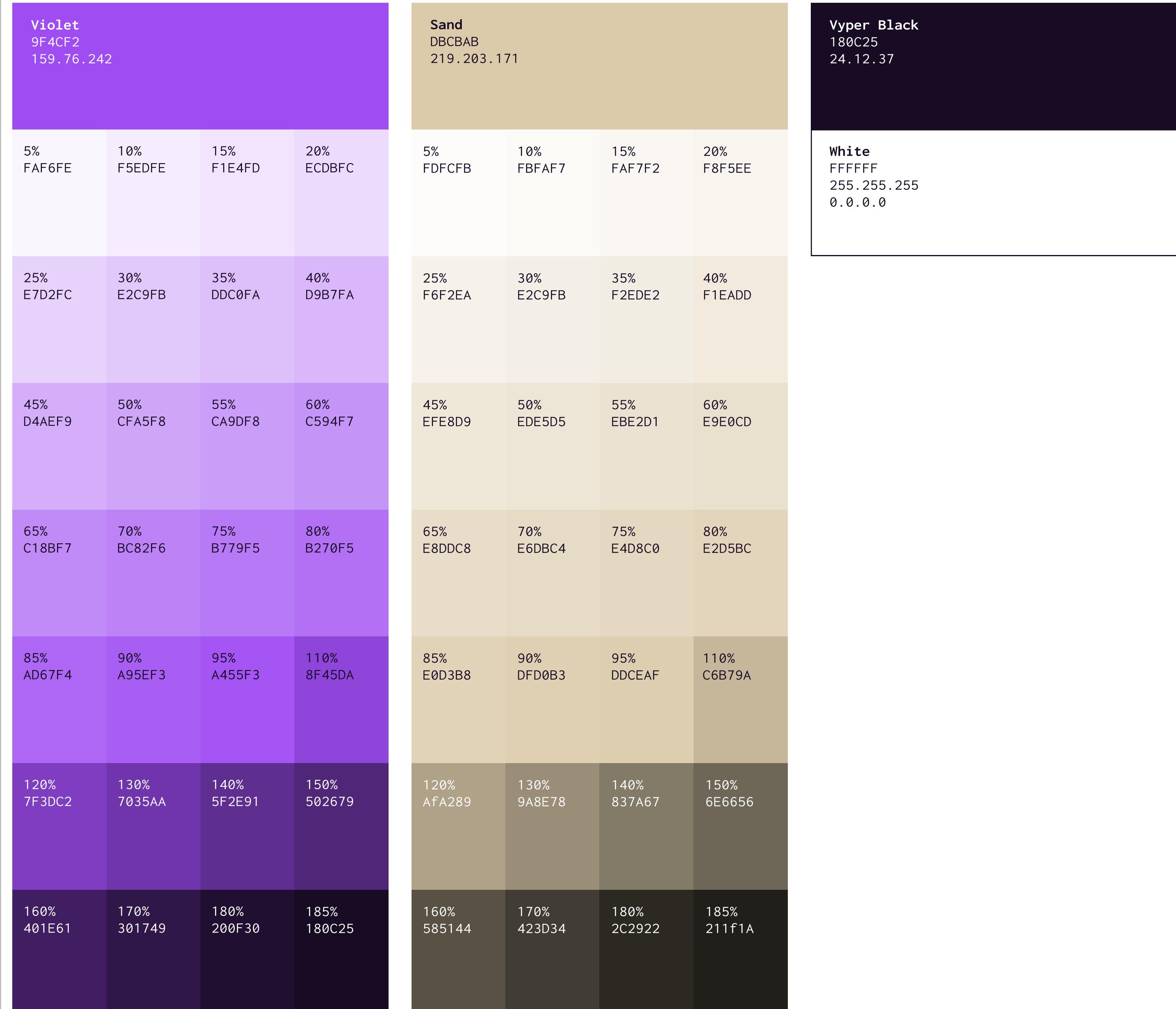
**Headlines set in  
SemiExpanded Bold**

SemiExpanded Medium for Highlighted paragraphs  
SemiExpanded Regular for Body copy  
Regular for body copy in small areas

# Color

Our primary palette consists of Violet, Sand, Vyper black and White.

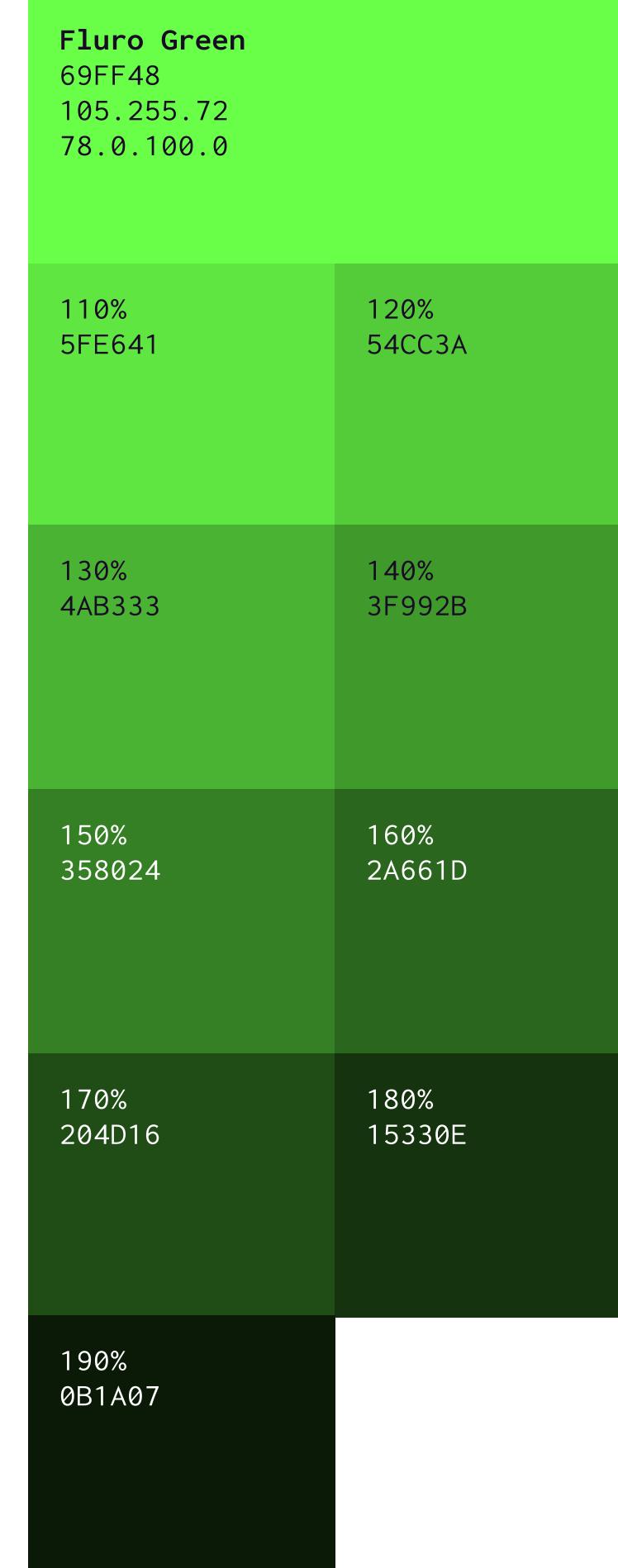
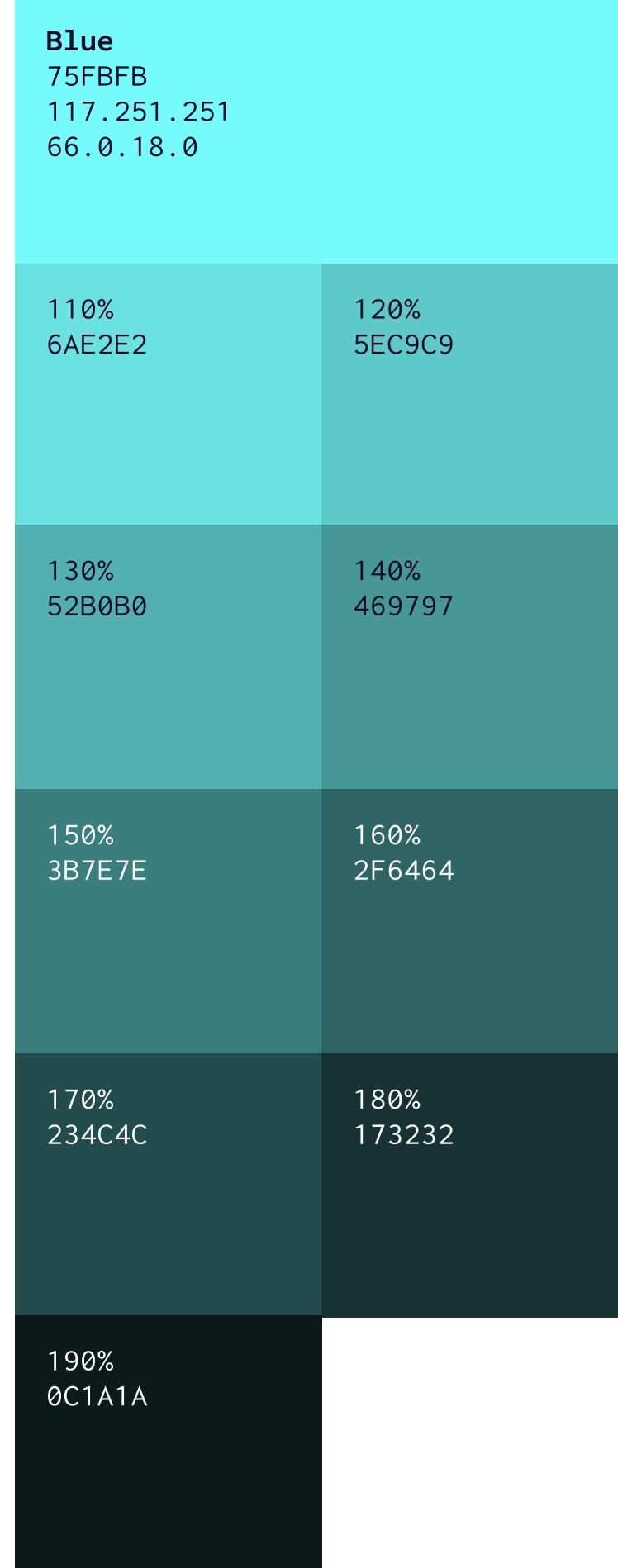
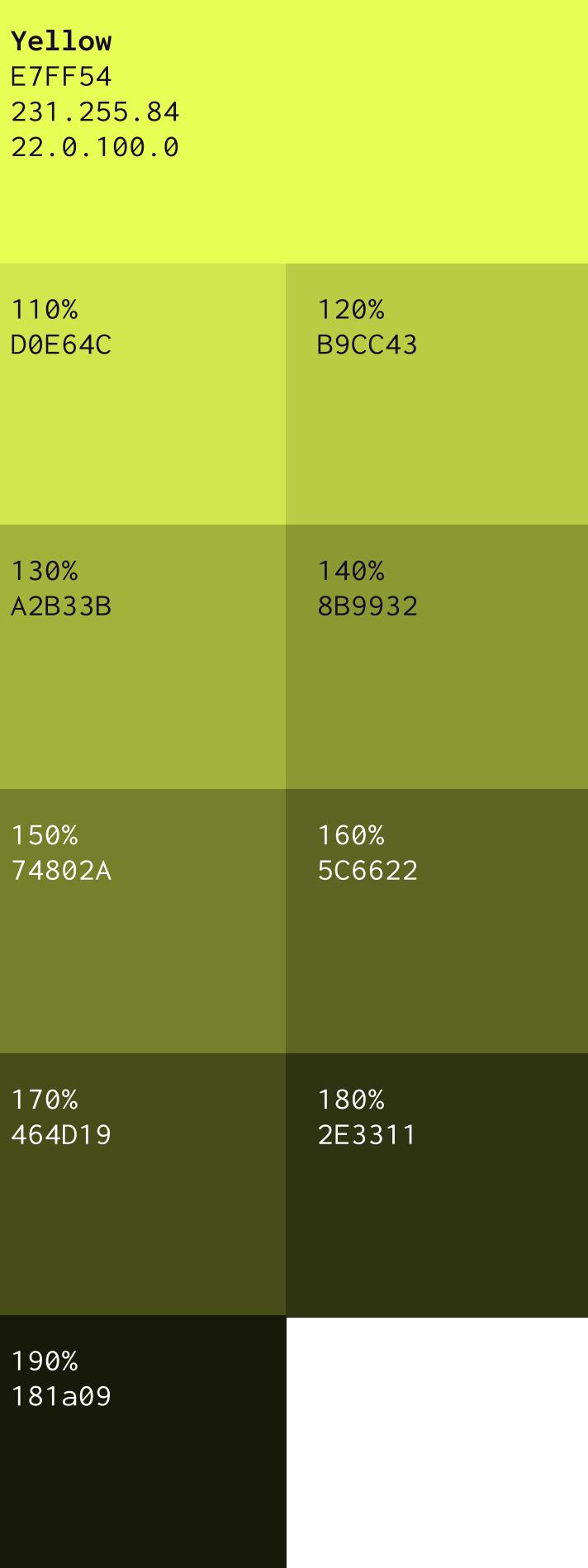
## Primary



# Color

Our secondary palette consists of Yellow, Blue, Emerald Green, Fluro Green and Orange.

## Secondary



# Using color

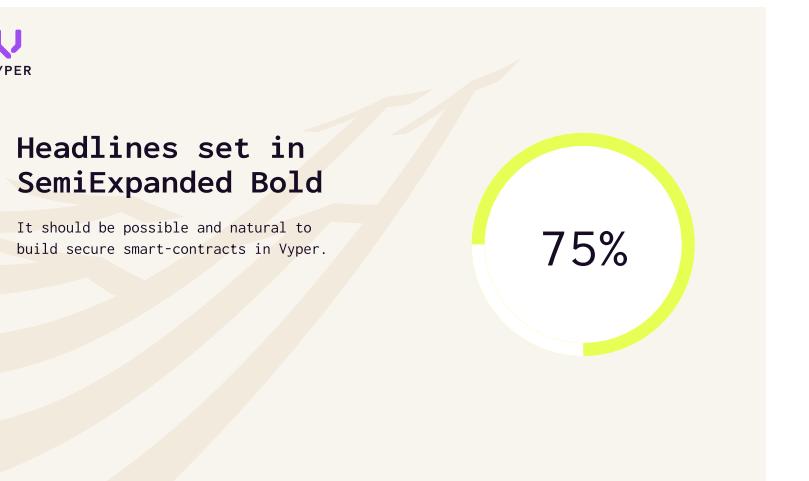
## Examples

Showing versatility  
in information graphics.

Light sand background  
with Violet highlights



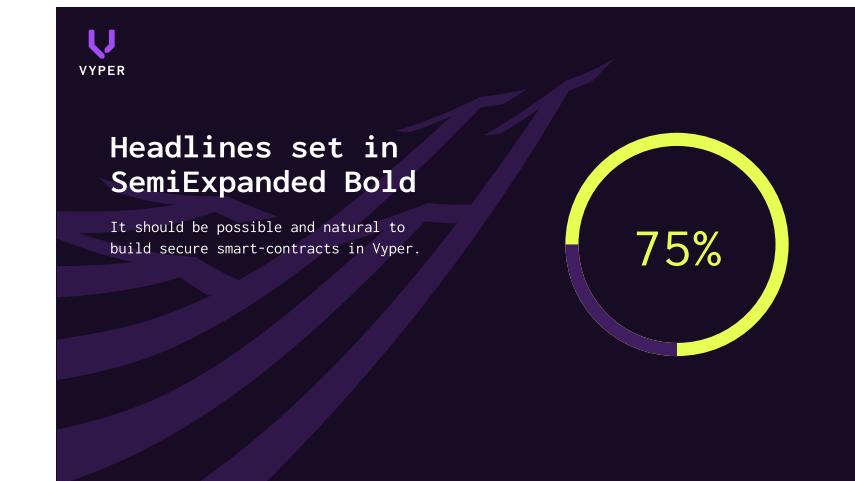
light sand background with secondary color highlights



id sand background with  
secondary color highlights



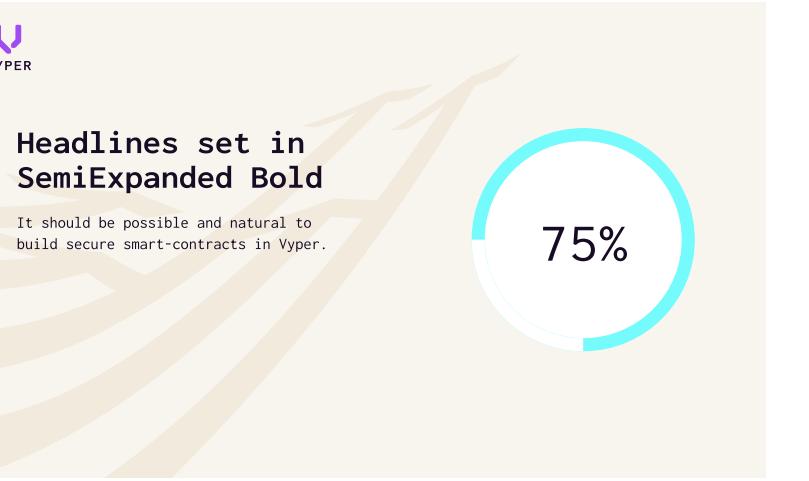
Vyper black background with secondary color highlights



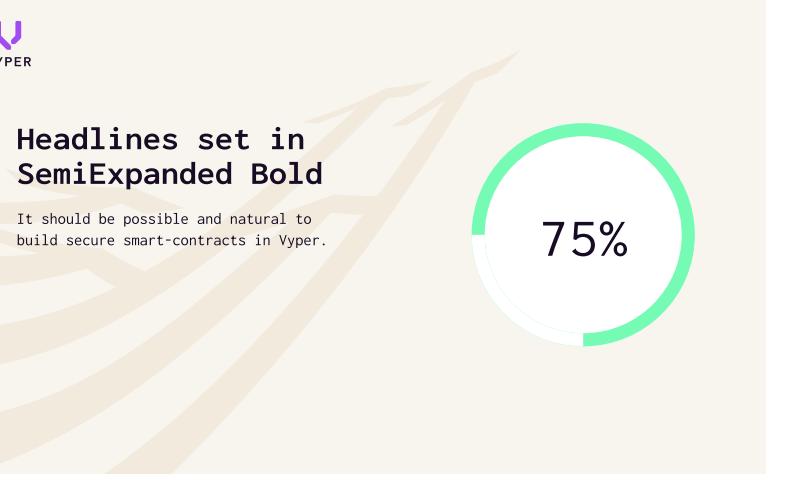
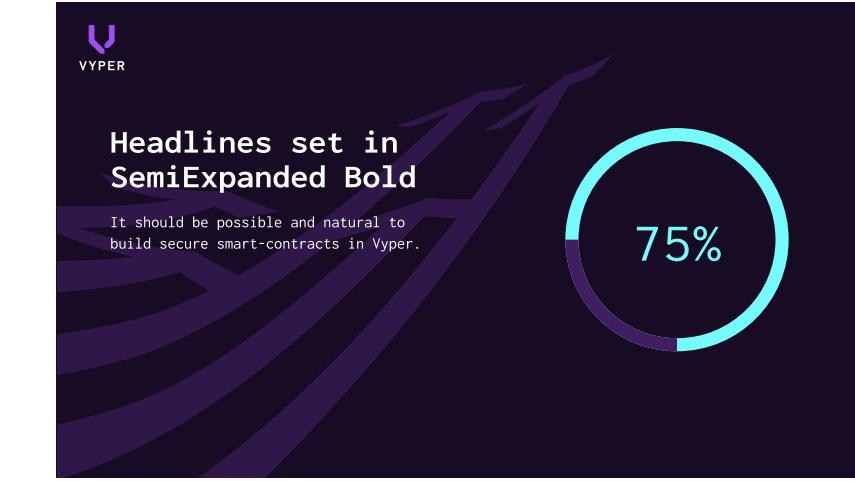
**Headlines set in  
SemiExpanded Bold**

It should be possible and natural to build  
secure smart-contracts in Vyper.

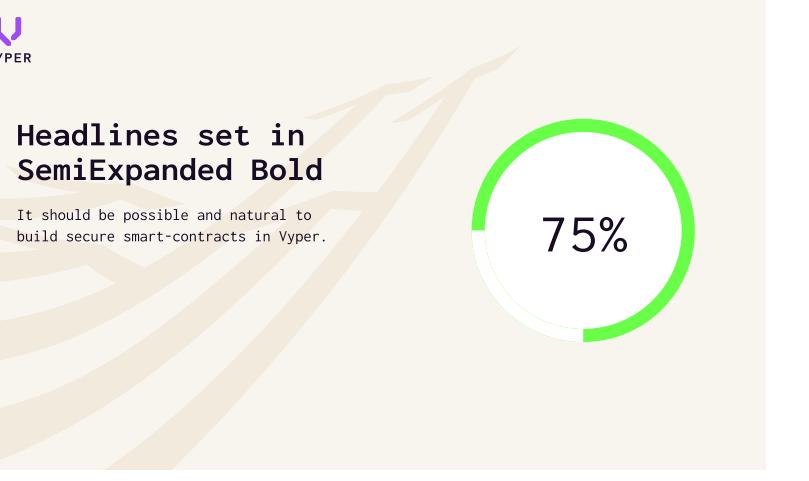
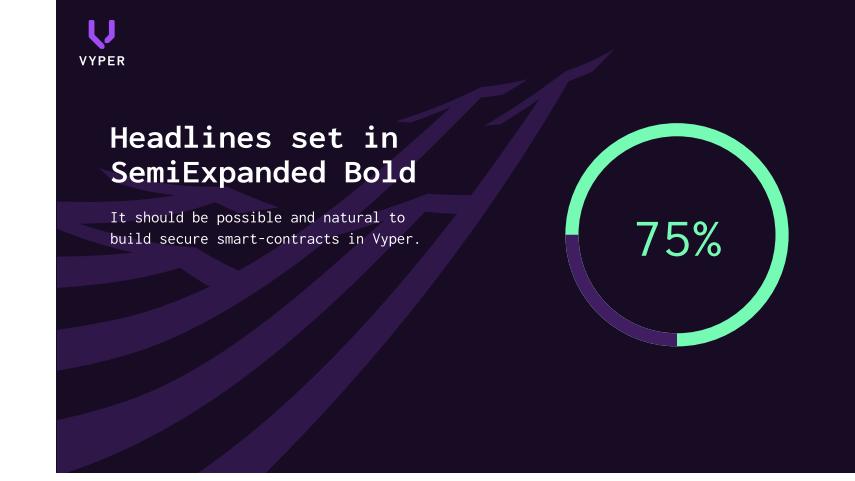
75%



The Vyper logo is located in the top-left corner. The word "VYPER" is written in a bold, black, sans-serif font below a stylized purple "V" icon. To the right of the logo, the text "Headlines set in SemiExpanded Bold" is displayed in a large, bold, black font. Below this, a smaller text block reads: "It should be possible and natural to build secure smart-contracts in Vyper." In the bottom-right corner, there is a large circular graphic with a light blue outline. The circle is divided into two segments: a white segment on the left containing the number "75%" and a light blue segment on the right.



The Vyper logo is located in the top-left corner. The main headline text, "Headlines set in SemiExpanded Bold", is displayed prominently in the center-left area. A large circular graphic on the right side contains the text "75%".

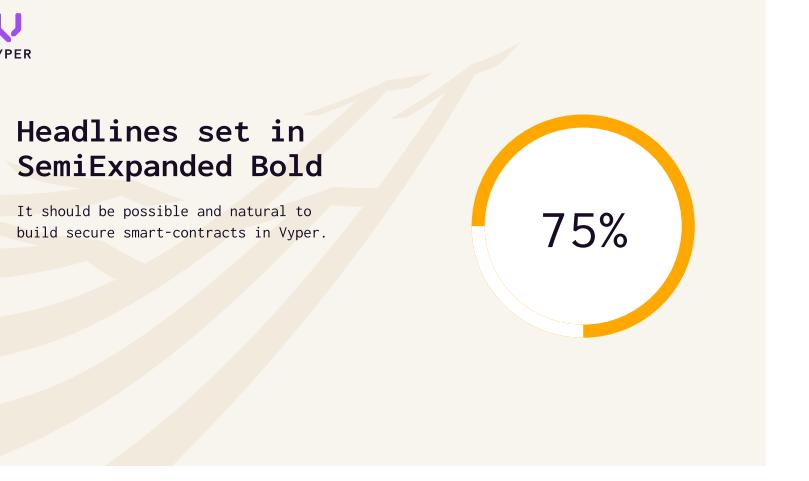
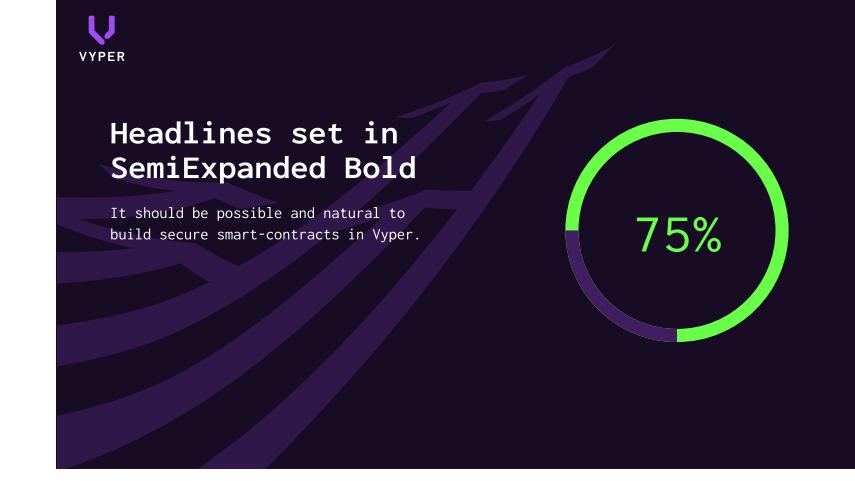


The Vyper logo is located in the top-left corner. The word "VYPER" is written in a bold, black, sans-serif font below the stylized purple "V" icon.

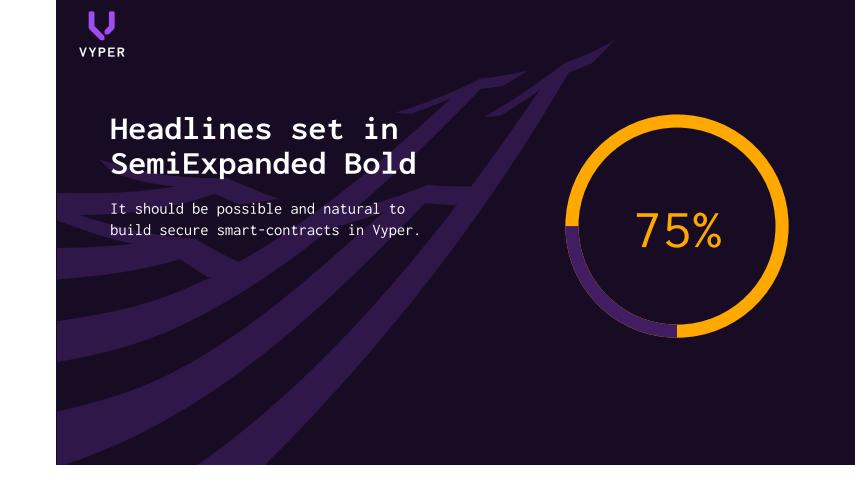
**Headlines set in  
SemiExpanded Bold**

It should be possible and natural to build secure smart-contracts in Vyper.

A large, semi-transparent circular progress bar is positioned on the right side of the slide. The bar has a thick green outline and is filled with white space. The number "75%" is displayed in the center of the circle in a large, bold, black font.



The slide features the Vyper logo at the top left. The main content area contains two sections: a headline in bold font and a descriptive text below it. To the right is a large circular progress bar indicating 75% completion.



Headlines using Violet with Body Copy and CTAs  
in Vyper Black on different tints of Sand backgrounds

## Using color

# Language

Language and compiler simplicity:  
The language and the compiler  
implementation should strive to  
be simple.

[more](#)

# Features

- Bounds and overflow checking: On arrays
- Support for signed integers and decimal numbers
- Decidability: It is possible to compute the result of any program
- Strong typing
- Small and understandable compiler core
- Limited support for pure functions:

Text editor  
using Sand and Secondary palette tints

```
1 # Events
2 event Transfer:
3     _from: indexed(address)
4     _to: indexed(address)
5     _value: uint256
6
7 event Approval:
8     _owner: indexed(address)
9     _spender: indexed(address)
10    _value: uint256
11
12 # Functions
13 @view
14 @external
15 def totalSupply() -> uint256:
16     pass
17
18 # Open Auction
19
20 # Auction params
21 # Beneficiary receives money from the highest bidder
22 beneficiary: public(address)
23 auctionStart: public(uint256)
24 auctionEnd: public(uint256)
25
26 # Current state of auction
27 highestBidder: public(address)
28 highestBid: public(uint256)
29
30 # Set to true at the end, disallows any change
ended: public(bool)
```

# Supergraphic

Our supergraphics are used primarily in a supporting capacity in key areas.

We have two versions which can be used in a number of variations and crops for different formats.

**Supergraphic No.1**  
Color and crops



**Supergraphic No.2**  
Color and crops



# Applications

## Marketing

Landing page

Supergraphic No.1

## Features

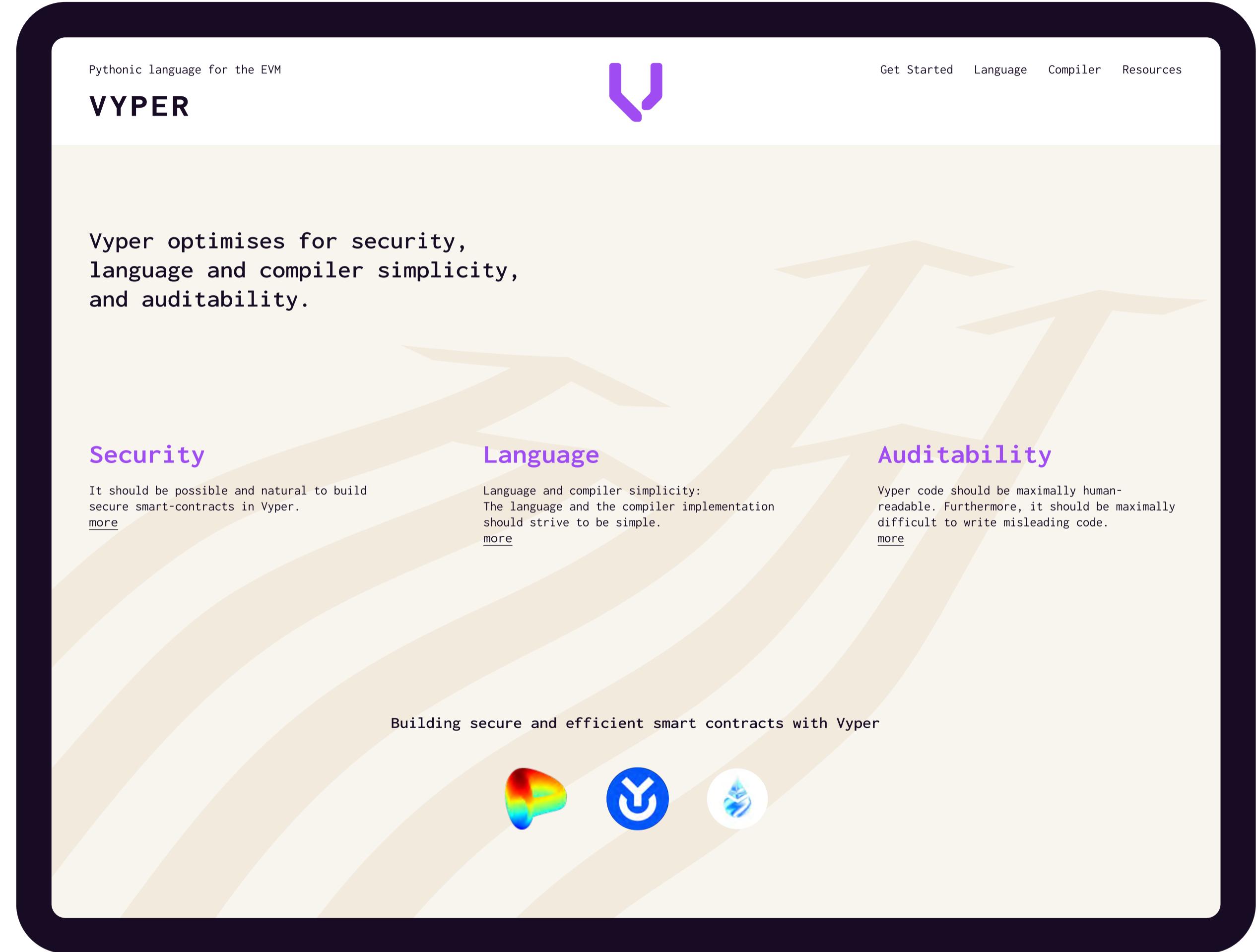
- Bounds and overflow checking: On array accesses and arithmetic.
- Support for signed integers and decimal fixed point numbers
- Decidability: It is possible to compute a precise upper bound for the gas consumption of any Vyper function call.
- Strong typing
- Small and understandable compiler code
- Limited support for pure functions: Anything marked constant is not allowed to change the state.

# Applications

## Marketing

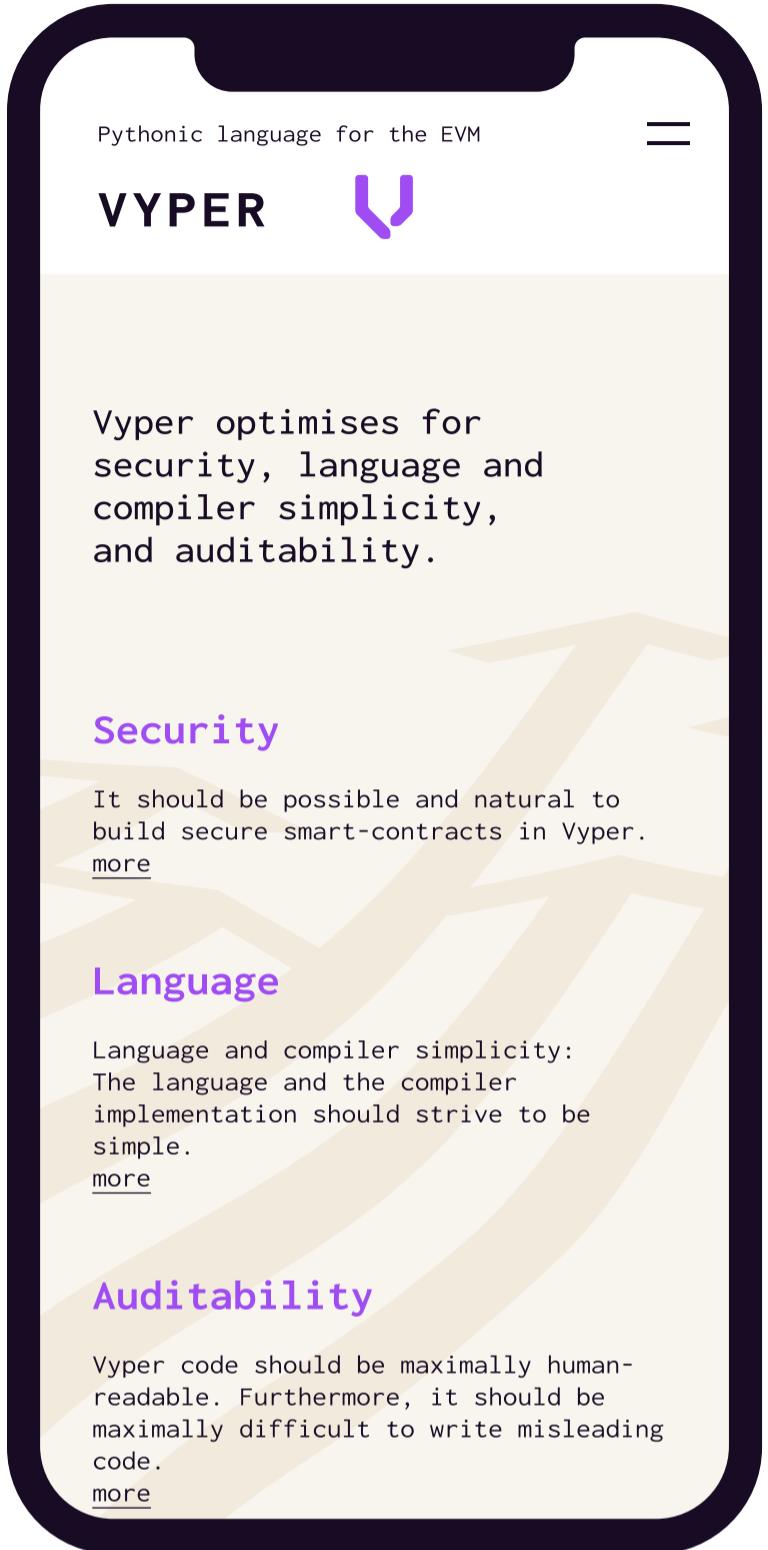
Landing page

Supergraphic No.2



## Features

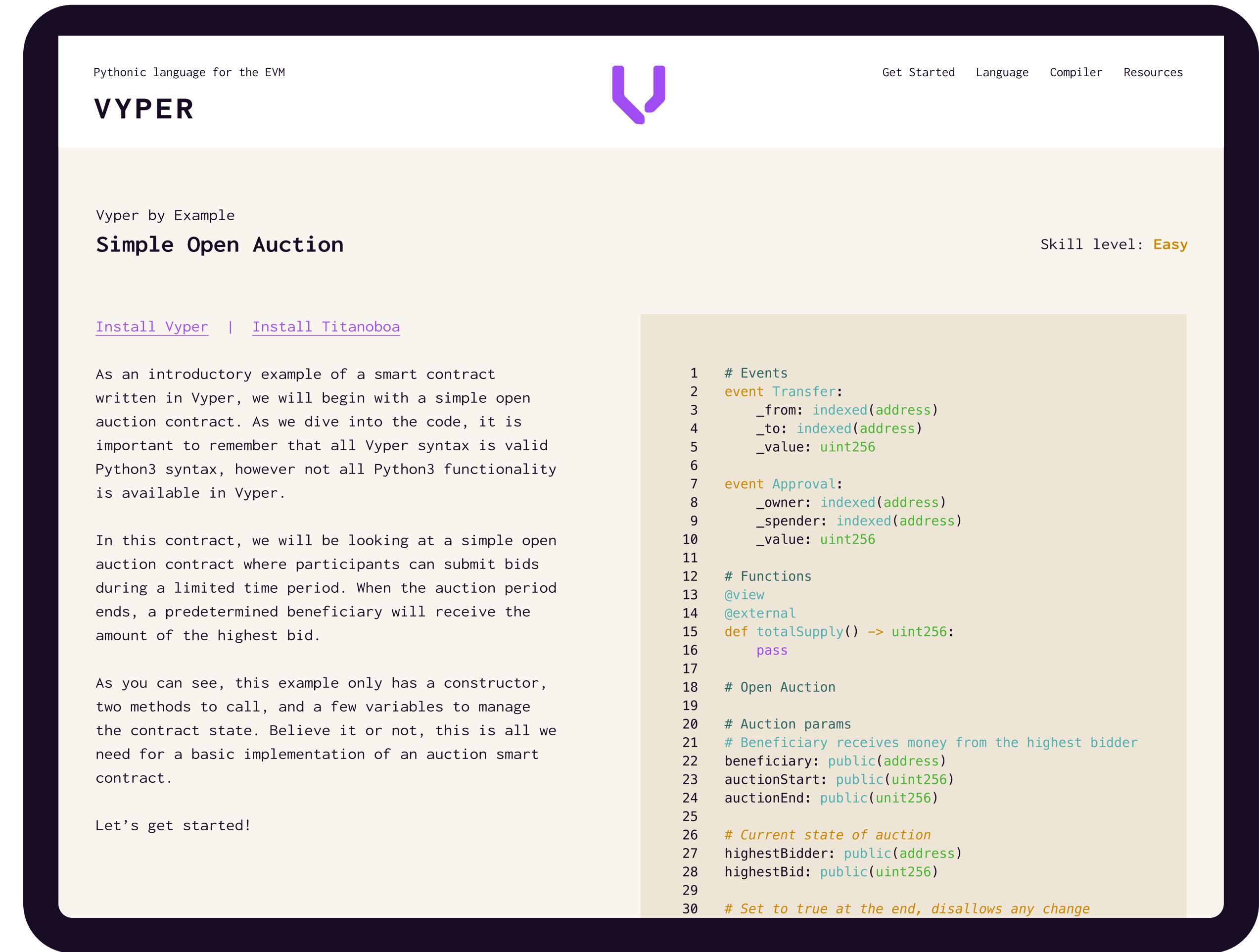
- Bounds and overflow checking: On array accesses and arithmetic.
- Support for signed integers and decimal fixed point numbers
- Decidability: It is possible to compute a precise upper bound for the gas consumption of any Vyper function call.
- Strong typing
- Small and understandable compiler code
- Limited support for pure functions: Anything marked constant is not allowed to change the state.



# Applications

## Developer

Landing page



The screenshot shows a dark-themed web page for the Vyper developer landing page. At the top, it says "Pythonic language for the EVM" and has a purple logo. A navigation bar includes "Get Started", "Language", "Compiler", and "Resources". Below the header, the title "Vyper by Example" and "Simple Open Auction" are displayed, along with a "Skill level: Easy" badge. Two download links, "Install Vyper" and "Install Titanoboa", are shown. The main content area contains explanatory text and a code listing. The explanatory text discusses the creation of a simple open auction contract, mentioning the use of Vyper syntax which is Python3-like but lacks some functionality. It also notes that the example only includes a constructor, two methods, and a few variables to manage the contract state. The code listing shows the Vyper source code for the auction contract, including event definitions for transfers and approvals, function definitions for totalSupply and auction logic, and variable declarations for beneficiary, auction start/end times, and highest bidder.

```

1 # Events
2 event Transfer:
3     _from: indexed(address)
4     _to: indexed(address)
5     _value: uint256
6
7 event Approval:
8     _owner: indexed(address)
9     _spender: indexed(address)
10    _value: uint256
11
12 # Functions
13 @view
14 @external
15 def totalSupply() -> uint256:
16     pass
17
18 # Open Auction
19
20 # Auction params
21 # Beneficiary receives money from the highest bidder
22 beneficiary: public(address)
23 auctionStart: public(uint256)
24 auctionEnd: public(uint256)
25
26 # Current state of auction
27 highestBidder: public(address)
28 highestBid: public(uint256)
29
30 # Set to true at the end, disallows any change

```

As you can see, this example only has a constructor, two methods to call, and a few variables to manage the contract state. Believe it or not, this is all we need for a basic implementation of an auction smart contract. Let's get started!

# Applications

Social



Vyper ✅ @vyperlang . Oct 4

vyper 0.3.10 is out! <https://github.com/vyperlang/vyper/releases/tag/v0.3.10...> for binaries, <https://pypi.org/project/vyper/0.3.10/> from pip



132

432

787

