VYPER
Visual Identity

Toolkit

Version 1.0

VYPER

VYPER
Visual Identity

Overview

Logo

VYPER

VYPER

VYPER

VYPER

Color

Violet	Violet 75%	Violet 50%	Violet 15%
9F4CF2	B779F5	CFA5F8	F1E4FD

Sand	Sand 75%	Sand 50%	Sand 15%
DBCBAB	E4D8C0	EDE5D5	FAF7F2

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

Positive

Negative

Mono portrait versions

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

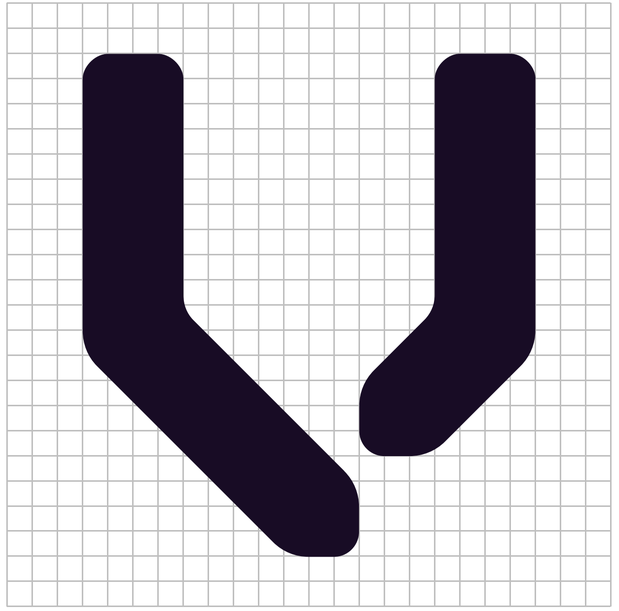
VYPER
Visual Identity

Symbol

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

Avatar

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

VYPER
Visual Identity

Color

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

Primary

Violet 9F4CF2 159.76.242 67.74.0.0			
5% FAF6FE	10% F5EDFE	15% F1E4FD	20% ECDBFC
25% E7D2FC	30% E2C9FB	35% DDC0FA	40% D9B7FA
45% D4AEF9	50% CFA5F8	55% CA9DF8	60% C594F7
65% C18BF7	70% BC82F6	75% B779F5	80% B270F5
85% AD67F4	90% A95EF3	95% A455F3	110% 8F45DA
120% 7F3DC2	130% 7035AA	140% 5F2E91	150% 502679
160% 401E61	170% 301749	180% 200F30	185% 180C25

Sand DBCBAB 219.203.171 16.18.36.2			
5% FDFCFB	10% FBFAF7	15% FAF7F2	20% F8F5EE
25% F6F2EA	30% E2C9FB	35% F2EDE2	40% F1EADD
45% EFE8D9	50% EDE5D5	55% EBE2D1	60% E9E0CD
65% E8DDC8	70% E6DBC4	75% E4D8C0	80% E2D5BC
85% E0D3B8	90% DFD0B3	95% DDCEAF	110% C6B79A
120% AfA289	130% 9A8E78	140% 837A67	150% 6E6656
160% 585144	170% 423D34	180% 2C2922	185% 211f1A

Vyper Black
200F30
32.15.48
95.100.44.63

White
FFFFFF
255.255.255
0.0.0.0

Color

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

Secondary

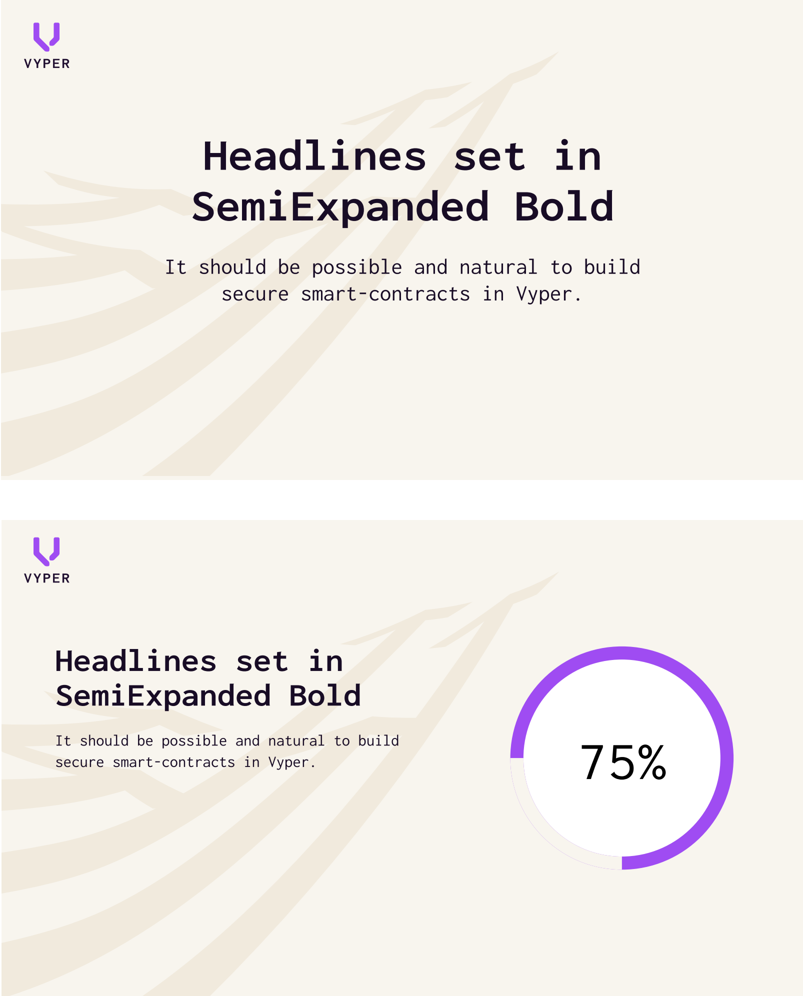
Yellow	Blue	Emerald Green	Fluro Green	Orange
E7FF54	75FBFB	75FBB4	69FF48	FFA800
231.255.84	117.251.251	117.251.180	105.255.72	255.168.0
22.0.100.0	66.0.18.0	68.0.52.0	78.0.100.0	0.43.93.0
110% D0E64C	110% 6AE2E2	110% 6AE2A2	110% 5FE641	110% E69800
120% B9CC43	120% 5EC9C9	120% 5EC990	120% 54CC3A	120% CC8600
130% A2B33B	130% 52B0B0	130% 52B07E	130% 4AB333	130% B37500
140% 8B9932	140% 469797	140% 46976C	140% 3F992B	140% 996400
150% 74802A	150% 3B7E7E	150% 3B7E5A	150% 358024	150% 4D3300
160% 5C6622	160% 2F6464	160% 2F6448	160% 2A661D	160% 664300
170% 464D19	170% 234C4C	170% 234C36	170% 204D16	170% 4D3300
180% 2E3311	180% 173232	180% 173224	180% 15330E	180% 332200
190% 181a09	190% 0C1A1A	190% 0C1A12	190% 0B1A07	190% 1A1100

VYPER
Visual Identity

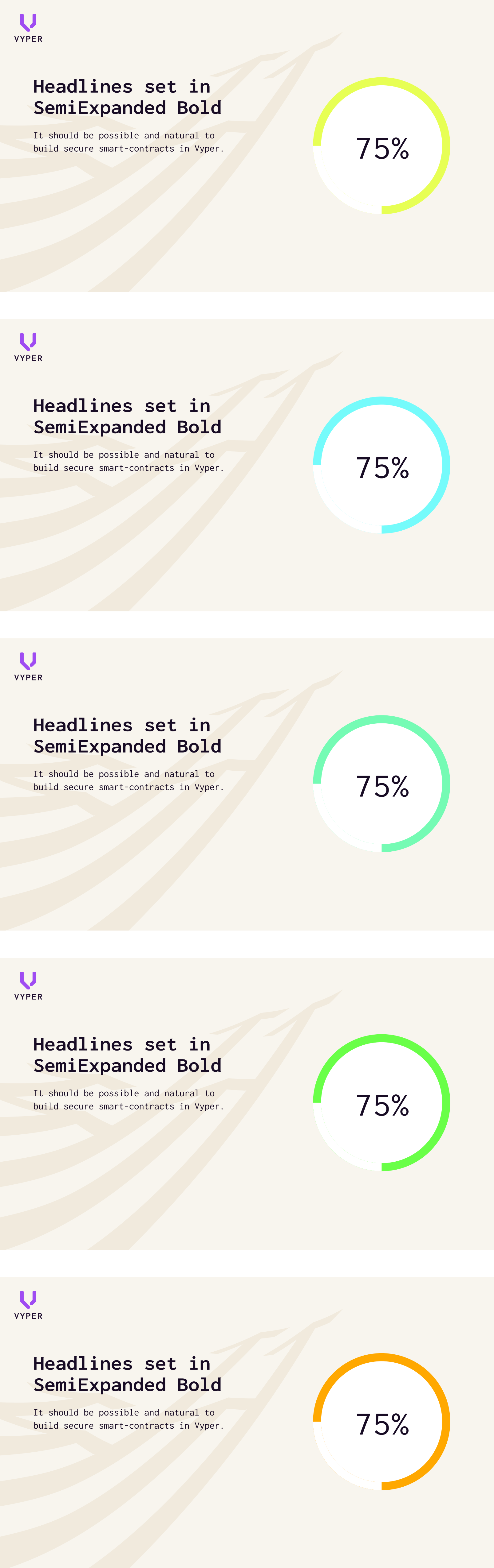
Using color

Examples
Showing versatility
in information graphics.

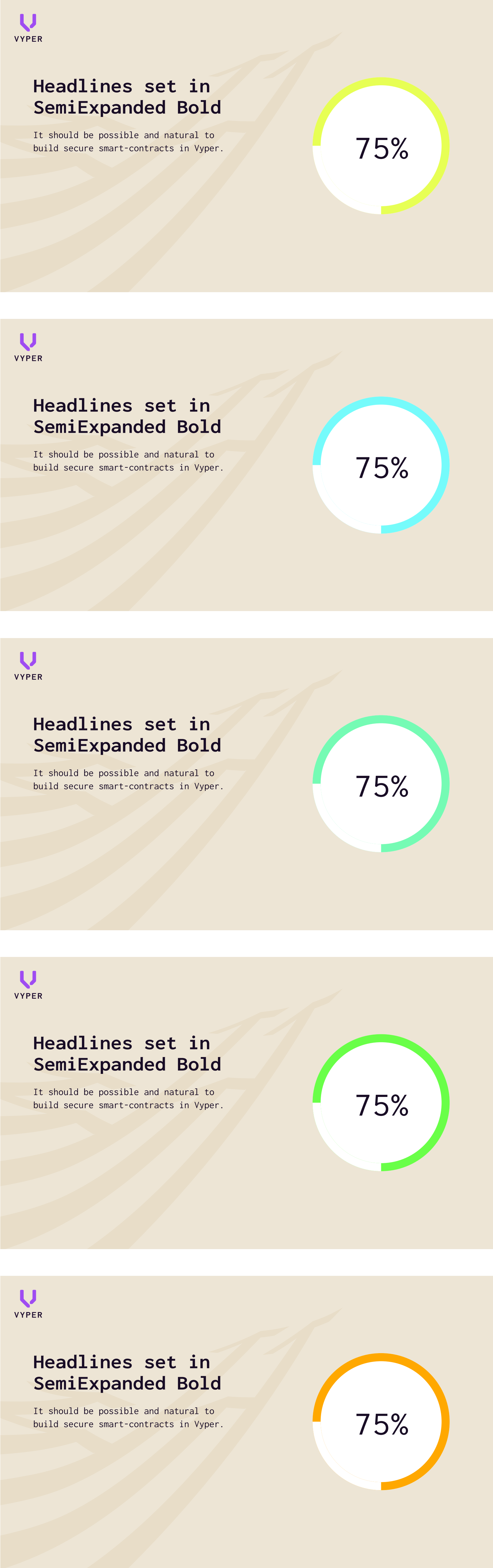
Light sand background
with Violet highlights



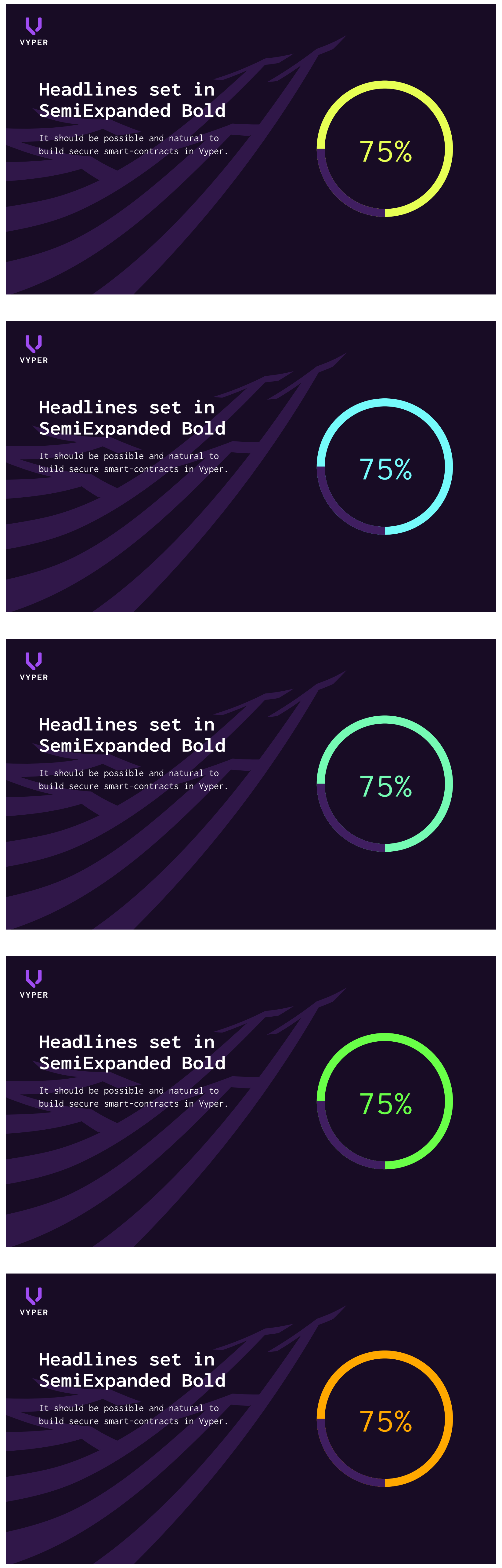
Light sand background with
secondary color highlights



Mid sand background with
secondary color highlights



Vyper black background with
secondary color highlights



Using color

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

Language

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

Features

- Bounds and overflow checking: On ar
- Support for signed integers and dec
- Decidability: It is possible to com
- Strong typing
- Small and understandable compiler c
- Limited support for pure functions:

Text editor
using Sand and Secondary palette tints

```
# Events
event Transfer:
    _from: indexed(address)
    _to: indexed(address)
    _value: uint256

event Approval:
    _owner: indexed(address)
    _spender: indexed(address)
    _value: uint256

# Functions
@view
@external
def totalSupply() -> uint256:
    pass

# Open Auction

# Auction params
# Beneficiary receives money from the highest bidder
beneficiary: public(address)
auctionStart: public(uint256)
auctionEnd: public(uint256)

# Current state of auction
highestBidder: public(address)
highestBid: public(uint256)

# 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

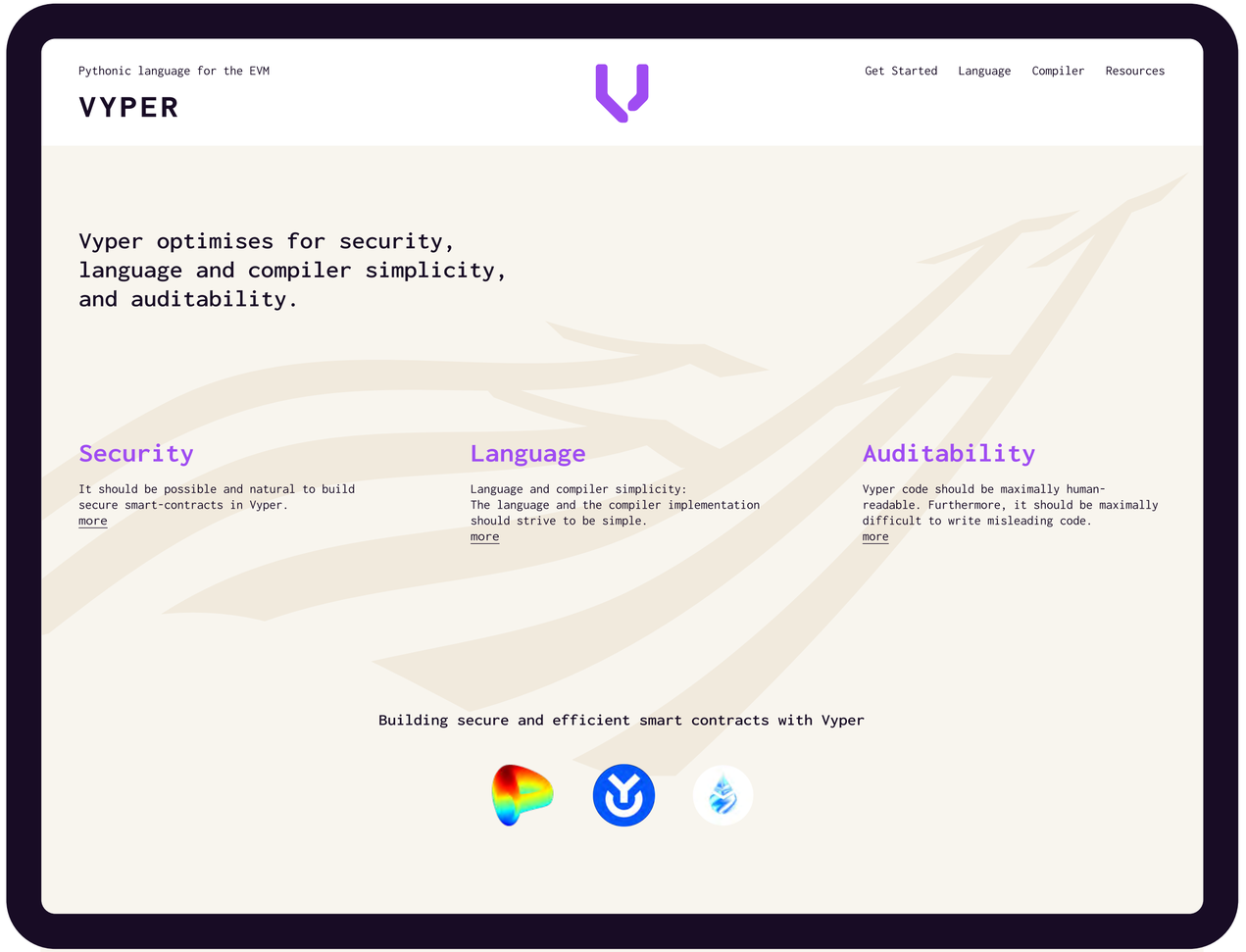
Background / Sand 20%
Supergraphic / Sand 40%

Supergraphic No.2
Color and crops

Background / Sand 20%
Supergraphic / Sand 40%

Applications

Marketing
Landing page
Supergraphic No.1

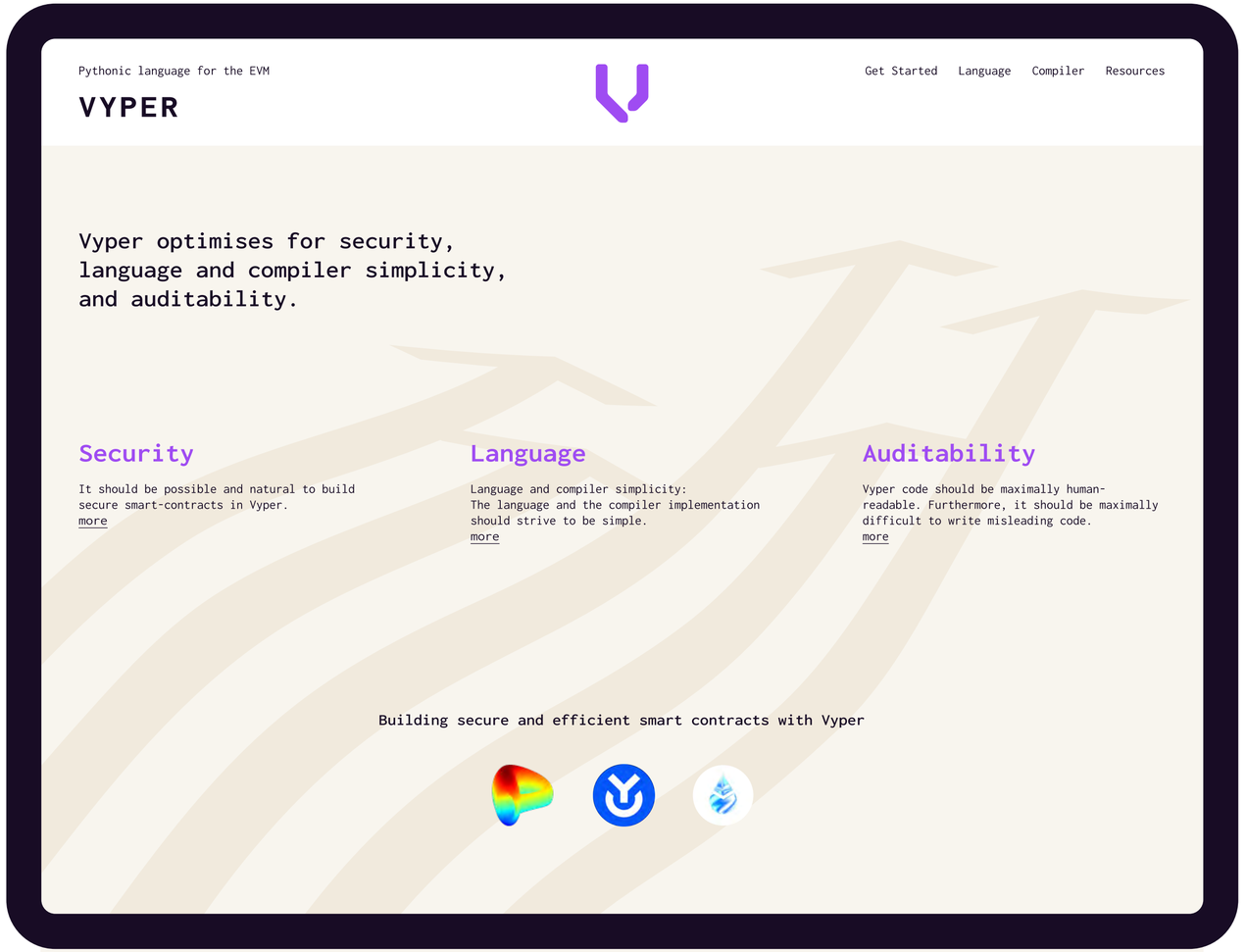


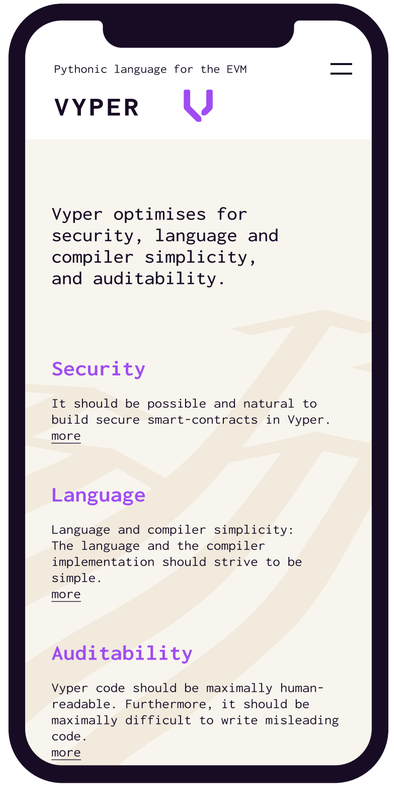
Features

- 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



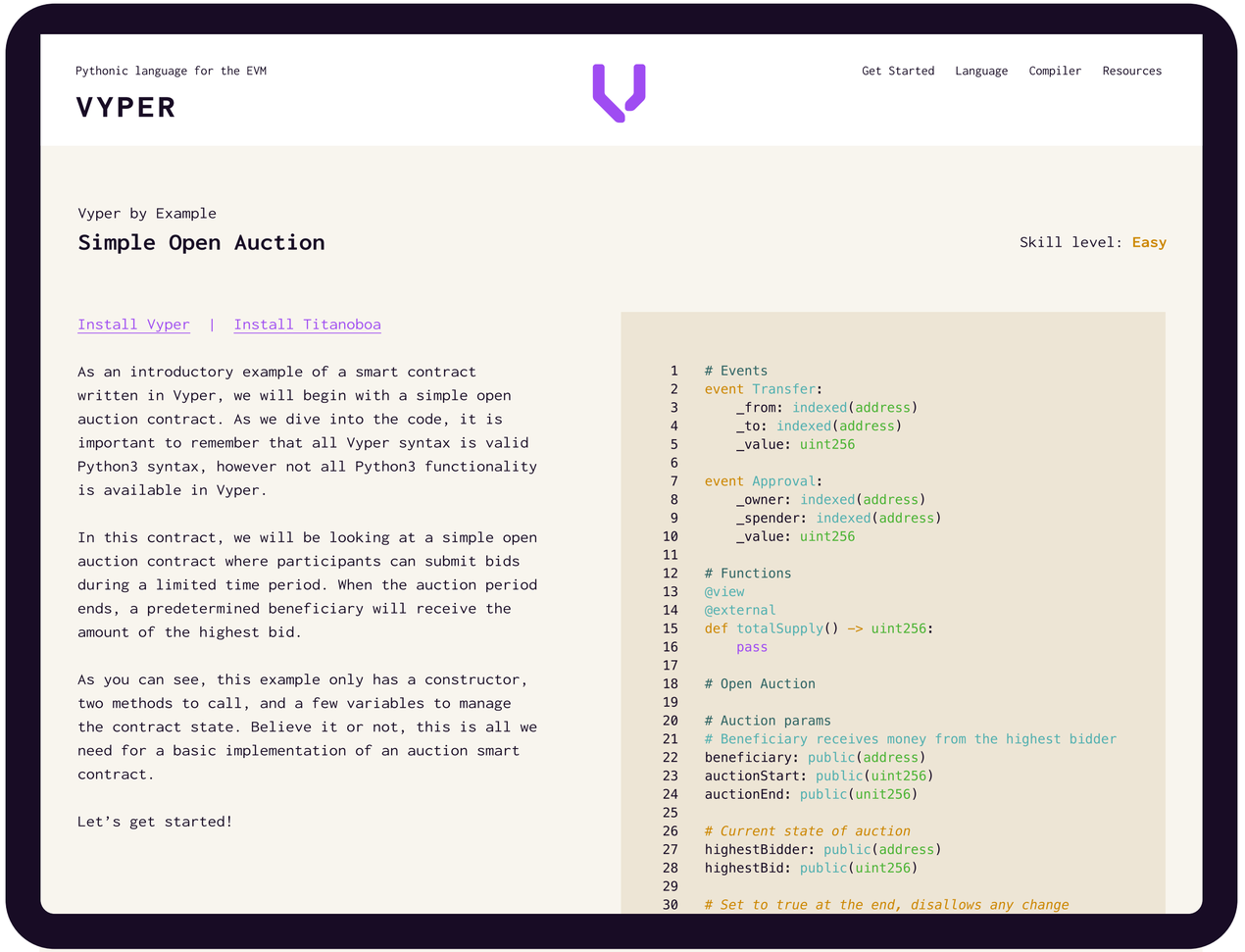


Features

- 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



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