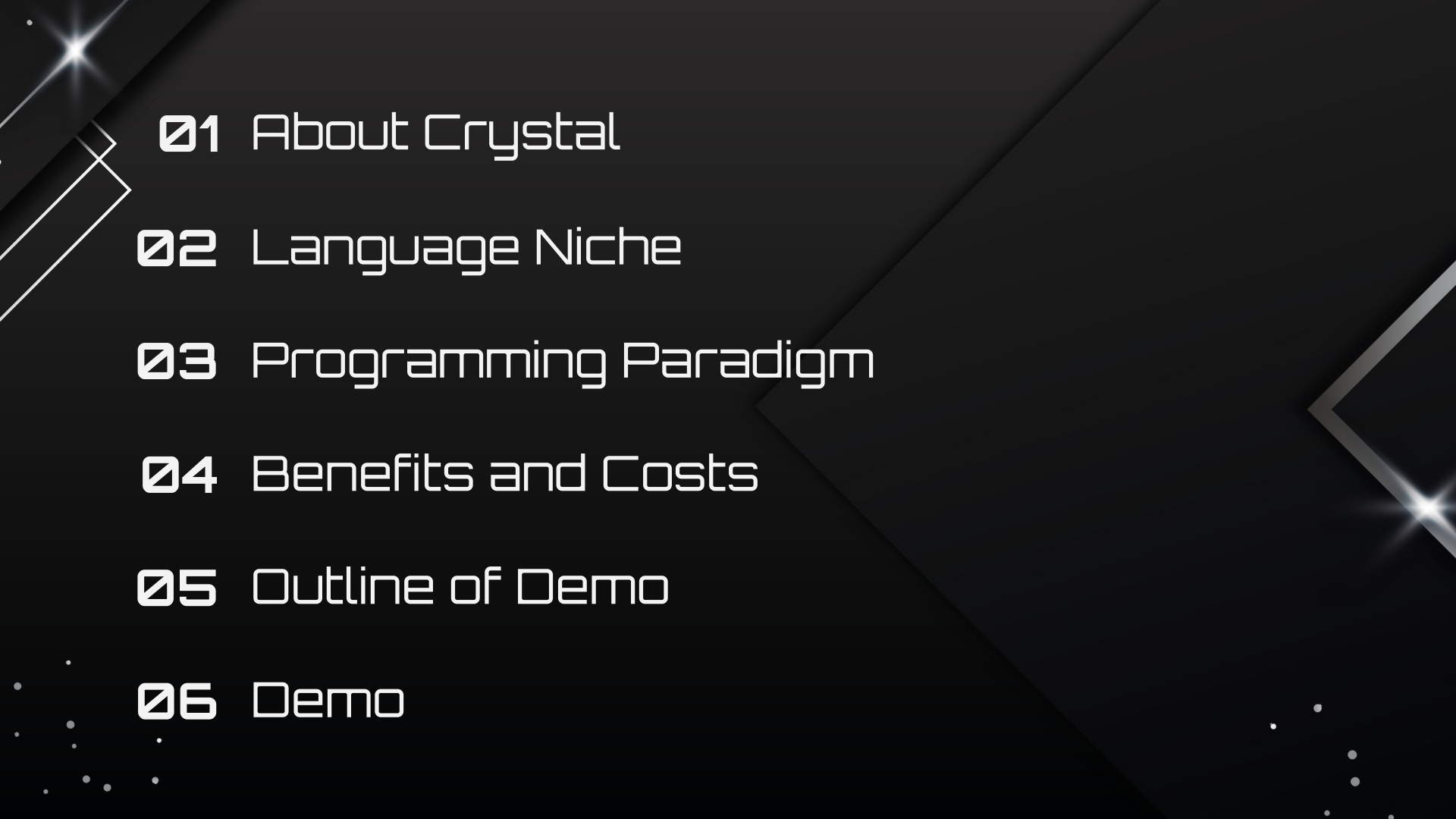




Crystal

Lucius Karki & Ryan Jackson

- 
- 01 About Crystal
 - 02 Language Niche
 - 03 Programming Paradigm
 - 04 Benefits and Costs
 - 05 Outline of Demo
 - 06 Demo



01

About Crystal

Creators



◆ Ary Borenszweig

Software Engineer

NoRedInk



◆ Juan Wajnerman

Full Stack Engineer

NoRedInk



◆ Brian Cardiff

Software Engineer

NoRedInk

All have previously worked at Manas Technology. All have previous experience with Ruby

Past

- ◇ Concept was created in June of 2011 under the name “Joy”
- ◇ First public release in 2014
- ◇ 4000 stars in December 2015
- ◇ Hit 32nd on the TIOBE index in 2017
- ◇ 1.0 was released in March of 2021

Present

- ◇ 19.7k stars on GitHub
- ◇ 1.6k forks
- ◇ Two main corporate sponsors:
 
- ◇ Hundreds of smaller/individual sponsors



Why?

We love Ruby's efficiency for writing code.

We love C's efficiency for running code.

We want the best of both worlds.

We want the compiler to understand what we mean without having to specify types everywhere.

We want full OOP.

Oh, and we don't want to write C code to make the code run faster.

It's Ruby but fast.

execution type	time (in seconds)
ruby	11.818
crystal run	02.371
crystal build	01.811
crystal build --release	00.303



02

Language Niche

Notable Features:

Shards

Libraries dependency manager.

C-Bindings

Allowed integration of preexisting c libraries.

Macros

Interpret AST nodes;
copy / paste code snippets

Concurrency

Uses green threads and channels, much like go.

What are the languages main competitors?





03

Programming Paradigm & Syntax

Language facts!

Statically Typed

Has type inference!

AOT Compiled

Unlike Ruby which is interpreted.

OOP

Although has
Functional features.

Garbage Collected

Uses Boehm Garbage
Collector.

Used in production

PlaceOS

Fully developed in Crystal
and TypeScript

NIKOLA®

Server side development
and infotainment system

84 codes

Hosting servers

Niches

Hosting

Strong Concurrency

Introductory

Very expressive
syntax

APIs

Kemal and Amber

CLI

Tools/Scripts

Some standard
library and is fast



04

Benefits and Costs

Pros

- Very easy for Rubyists to get into
- Fast
- Strong Concurrency
- Feels dynamically typed (type inference)
- C-Bindings
- Methodical development
- Web Development
 - Frameworks similar to Rails & Sinatra

Cons

- Small user base
- Slow development
 - New features
 - Windows
- “Washed up”
- Not backed by a large company
- Documentation
- Garbage collected

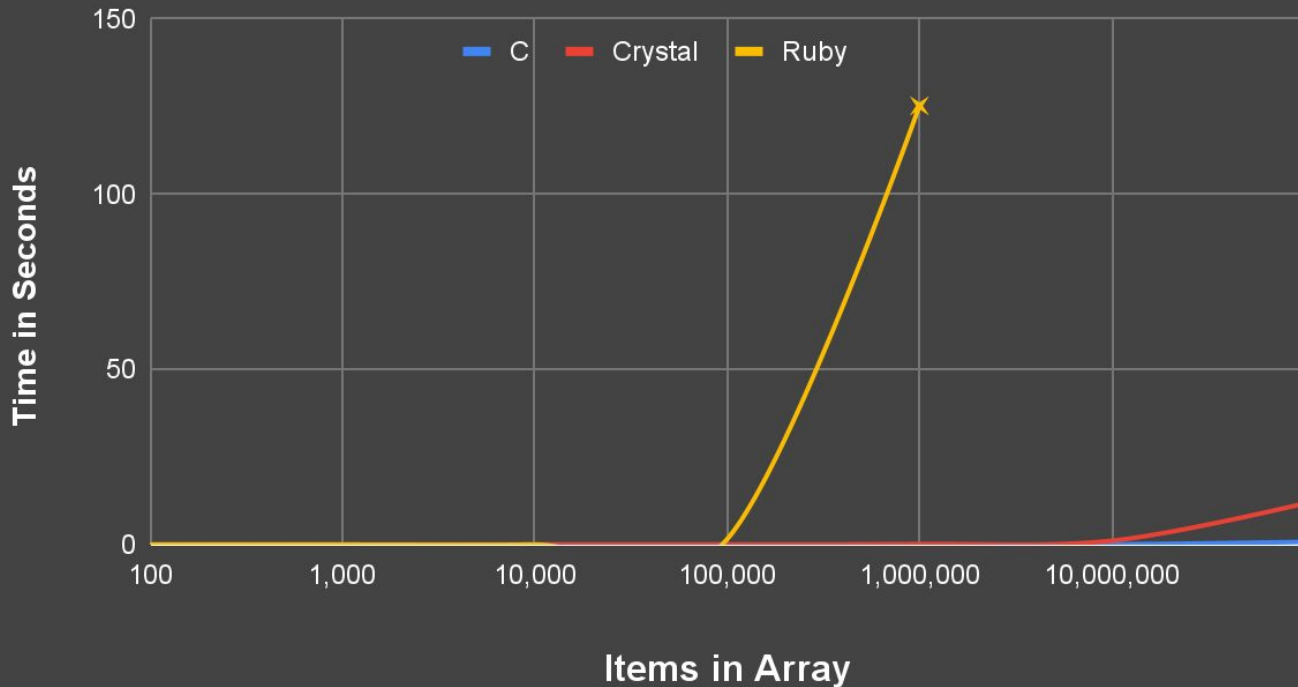
The background is a dark gray gradient. It features several white geometric elements: a large 'X' shape formed by multiple parallel lines in the top right and bottom right corners, and a large 'Z' or 'N' shape in the bottom left corner. Small, bright white starbursts are placed at the intersections of these lines. Scattered throughout the background are numerous small, faint white dots, resembling a starry sky or digital noise.

05

Outline of Demo

Demo Performance

Language Performance Comparison



The background is a dark gray gradient. It features several white geometric elements: a large 'X' shape formed by multiple parallel lines in the top right and bottom right corners; a single line forming a 'Z' shape in the bottom left corner; and two bright starburst light effects at the intersections of these line sets. Scattered white dots of varying sizes are also present, particularly in the top left and bottom center areas.

06

Demo



Questions?



Thanks!