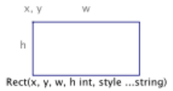
Workshop

Scalable Vector Graphics

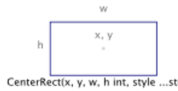
`io.Writer`

SVG Go Library

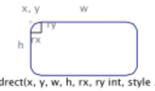
github.com/ajstarks/svgo



Rect(x, y, w, h int, style ...string)



CenterRect(x, y, w, h int, style ...string)

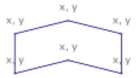


Roundrect(x, y, w, h, rx, ry int, style ...string)

Square(x, y, w int, style ...string)

Line(x1, y1, x2, y2 int, style ...string)

Polyline(x, y []int, style ...string)

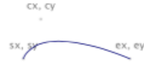


Polygon(x, y []int, style ...string)

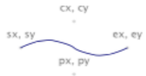
Circle(x, y, r int, style ...string)

Ellipse(x, y, rx, ry int, style ...string)

Arc(sx, sy, ax, ay, r int, lflag, sflag bool, ex, ey int, style ...string)

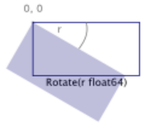


Qbez(sx, sy, cx, cy, ex, ey int, style ...string)

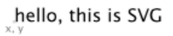


Bezier(sx, sy, cx, cy, px, py, ex, ey int, style ...string)

$ go get github.com/ajstarks/svgo



Rotate(r float64)



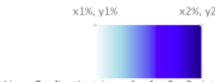
Text(x, y int, s string, style ...string)

Textpath(s, pathid string, style ...string)

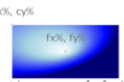
Path(s string, style ...string)

Image(x, y, w, h, int path string, style ...string)

Grid(x, y, w, h, n int, style ...string)

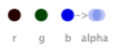


LinearGradient(s string, x1, y1, x2, y2 uint8, oc []Offcolor)



RadialGradient(s string, cx, cy, r, fx, fy uint8, oc []Offcolor)

RGB(r, g, b int)



RGBA(r, g, b int, opacity float64)

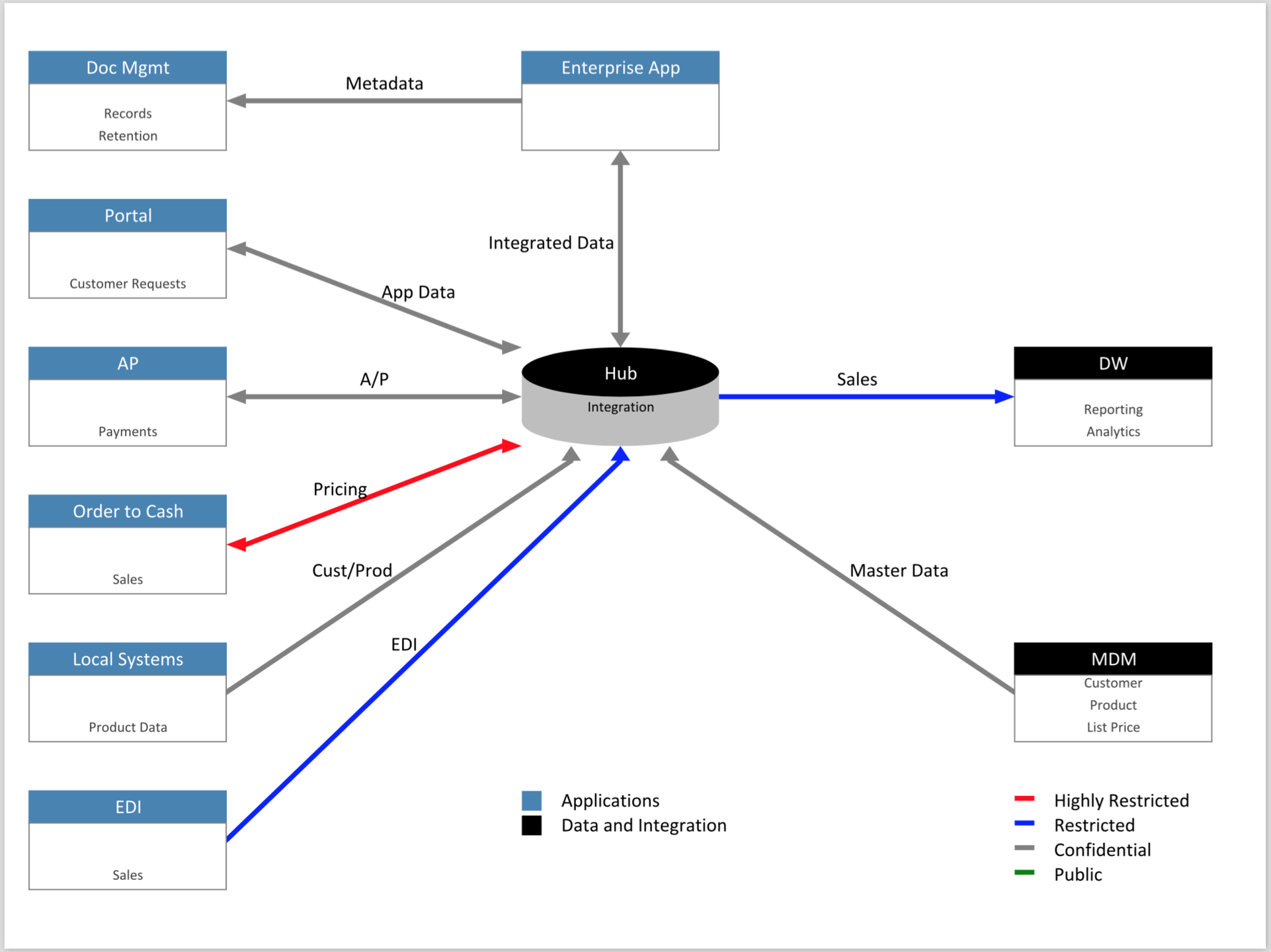
New(w io Writer)	specify destination
Start(w, h int, options ...string)/End()	begin/end the document
Startview(w, h, minx, miny, vw, vh int)	begin/end the document with viewport
Group(s ...string)/Gend()	begin/end group with attributes
Gstyle(s string)/Gend()	begin/end group style
Gtransform(s string)/Gend()	begin/end group transform
Gid(id string)/Gend()	begin/end group id
ClipPath(s ...string)/ClipEnd()	begin/end clip path
Def()/DefEnd()	begin/end a defintion block
Marker()/MarkerEnd()	begin/end markers
Pattern()/PatternEnd()	begin/end pattern
Desc(s string)	set the description element
Title(s string)	set the title element
Script(type, data ...string)	define a script
Mask(id string, x,y,w,h int, style ...string)/MaskEnd()	begin/end mask element
Link(href string, title string)/LinkEnd()	begin/end link to href, with a title
Use(x int, y int, link string, style ...string)	use defined objects

Textlines(x, y int, s []string, size, spacing int, fill, align string)

SVGo Clients in the Repo

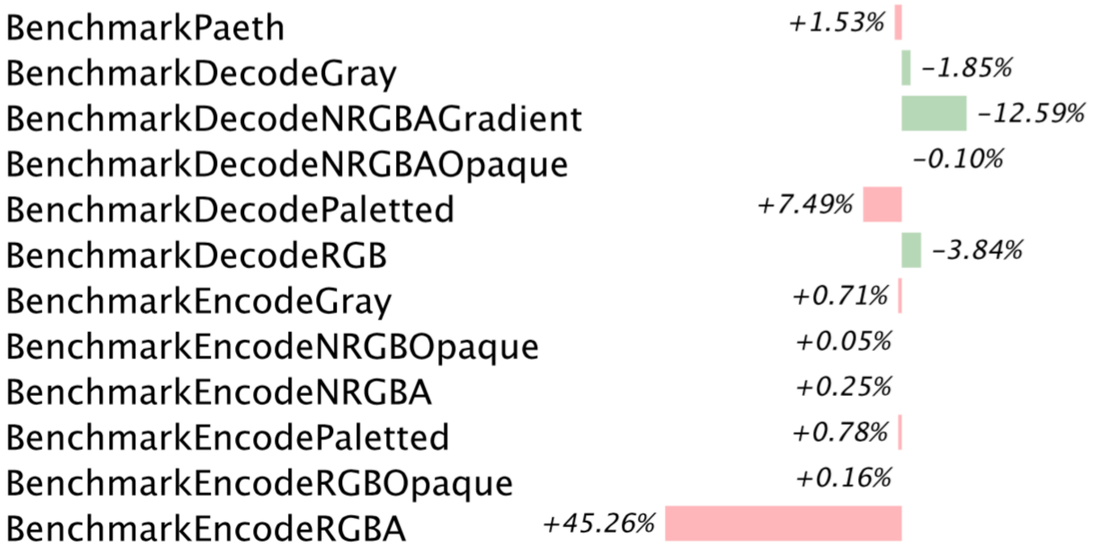
`svgdef`	Creates a SVG representation of the API
`android`	The Android logo
`bubtrail`	Bubble trails
`bulletgraph`	Bullet Graphs (via Stephen Few)
`colortab`	Display SVG named colors with RGB values
`compx`	Component diagrams
`flower`	Random flowers
`fontcompare`	Compare two fonts
`f50`	Get 50 photos from Flickr from a query
`fe`	Filter effects
`funnel`	Funnel from transparent circles
`gradient`	Linear and radial gradients
`html5logo`	HTML5 logo with draggable elements
`imfade`	Show image fading
`lewitt`	Version of Sol Lewitt's Wall Drawing 91
`ltr`	Layer Tennis Remixes
`marker`	Test markers
`paths`	Demonstrate SVG paths
`pattern`	Test patterns
`planets`	Show the scale of the Solar system
`pmap`	Proportion maps
`randcomp`	Compare random number generators
`richter`	Gerhard Richter's 256 colors
`rl`	Random lines
`skewabc`	Skew ABC
`stockproduct`	Visualize product and stock prices
`svgopher`	SVGo Mascot
`svgplay`	SVGo sketching server
`svgplot`	Plot data
`svgrid`	Compose SVG files in a grid
`tsg`	Twitter Search Grid
`tumblrgrid`	Tumblr picture grid
`turbulence`	Turbulence filter effect
`vismem`	Visualize data from files
`webfonts`	"Hello, World" with Google Web Fonts
`websvg`	Generate SVG as a web server

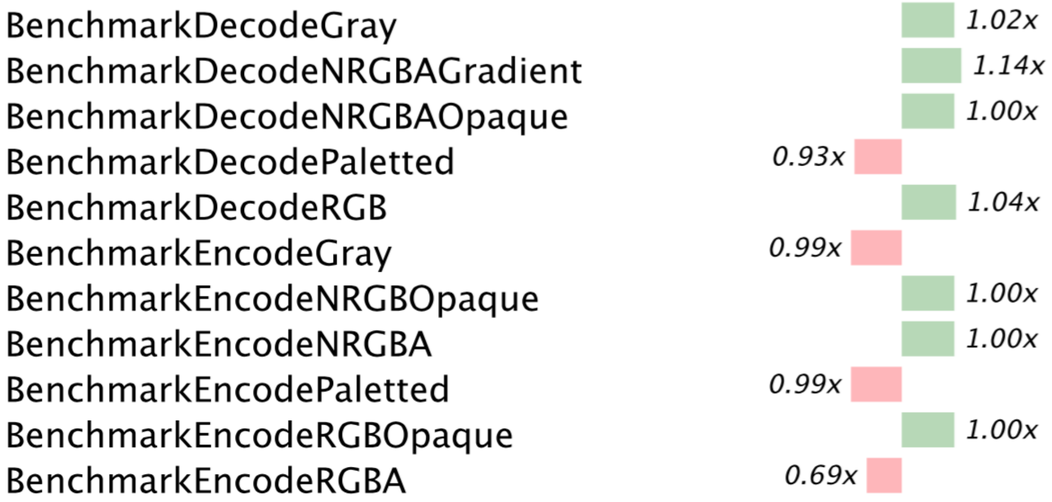
SVGo Clients

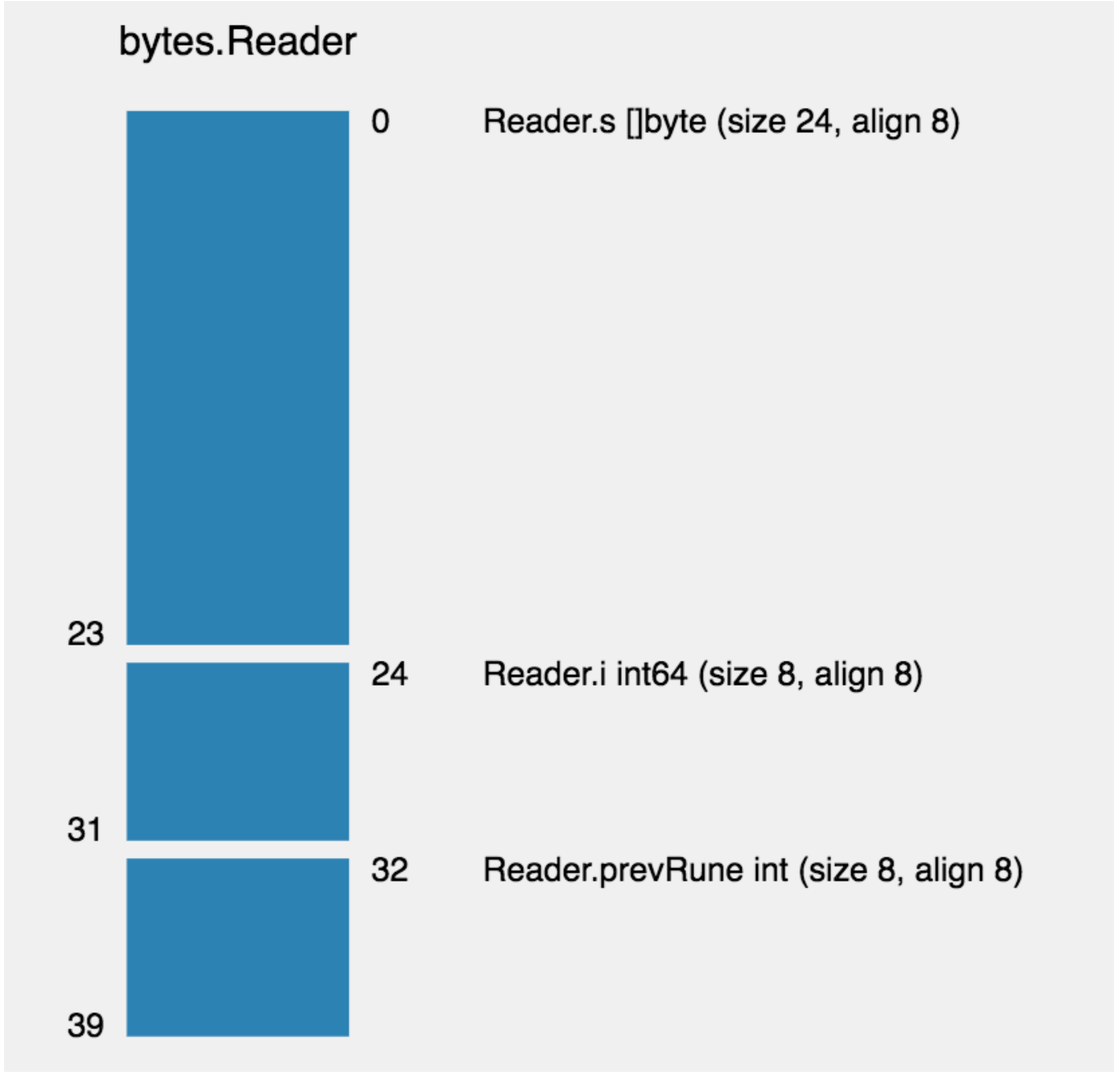

Doc Mgmt
Records
Retention
Metadata
Enterprise App
Portal
Customer Requests
App Data
Integrated Data
AP
Payments
A/P
Hub
Integration
Sales
DW
Reporting
Analytics
Pricing
Order to Cash
Sales
Cust/Prod
Master Data
Local Systems
Product Data
EDI
MDM
Customer
Product
List Price
EDI
Sales
Applications
Data and Integration
Highly Restricted
Restricted
Confidential
Public

benchpng.txt

image/png package: Go 1.3 vs Go 1.4

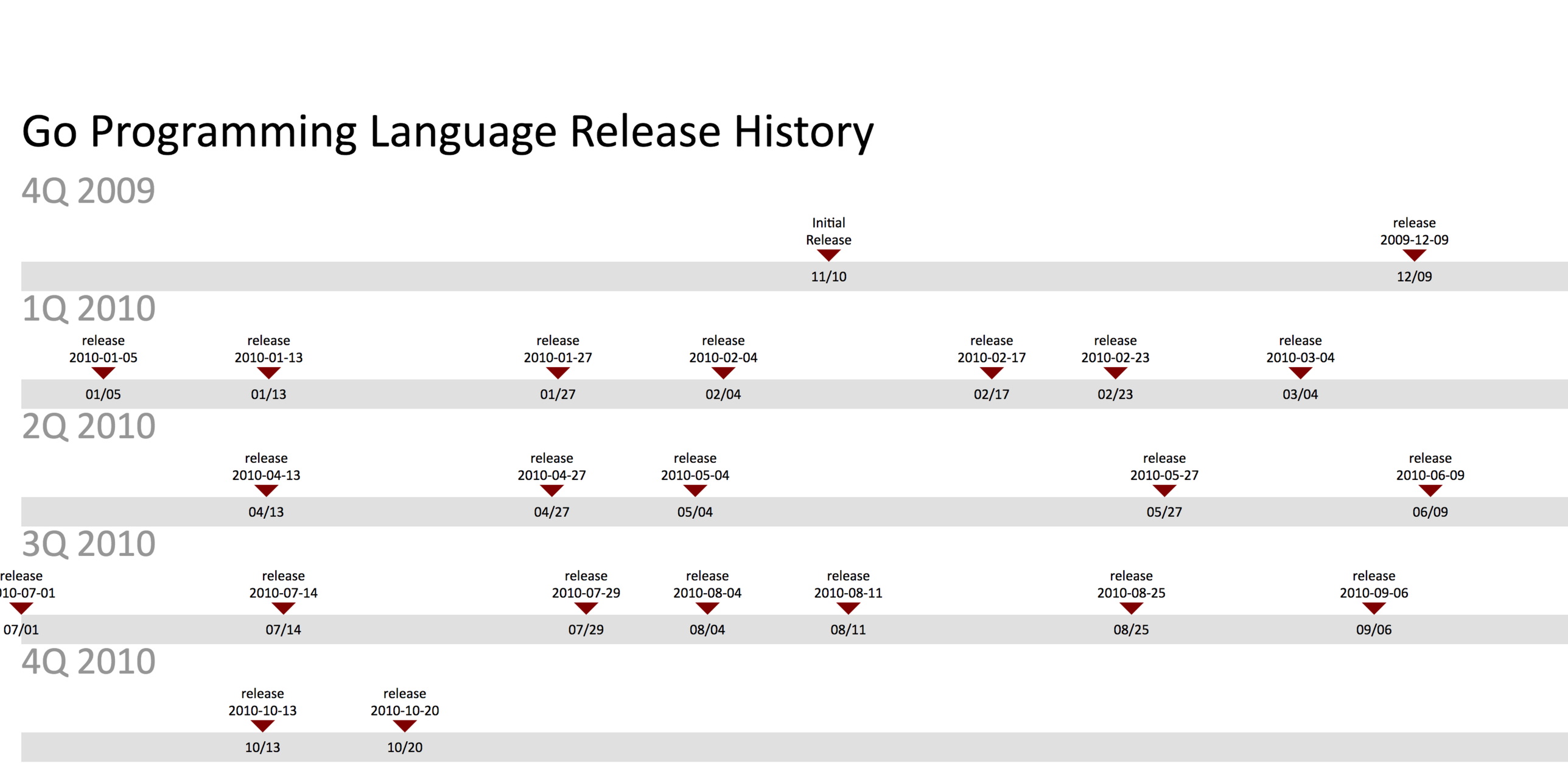


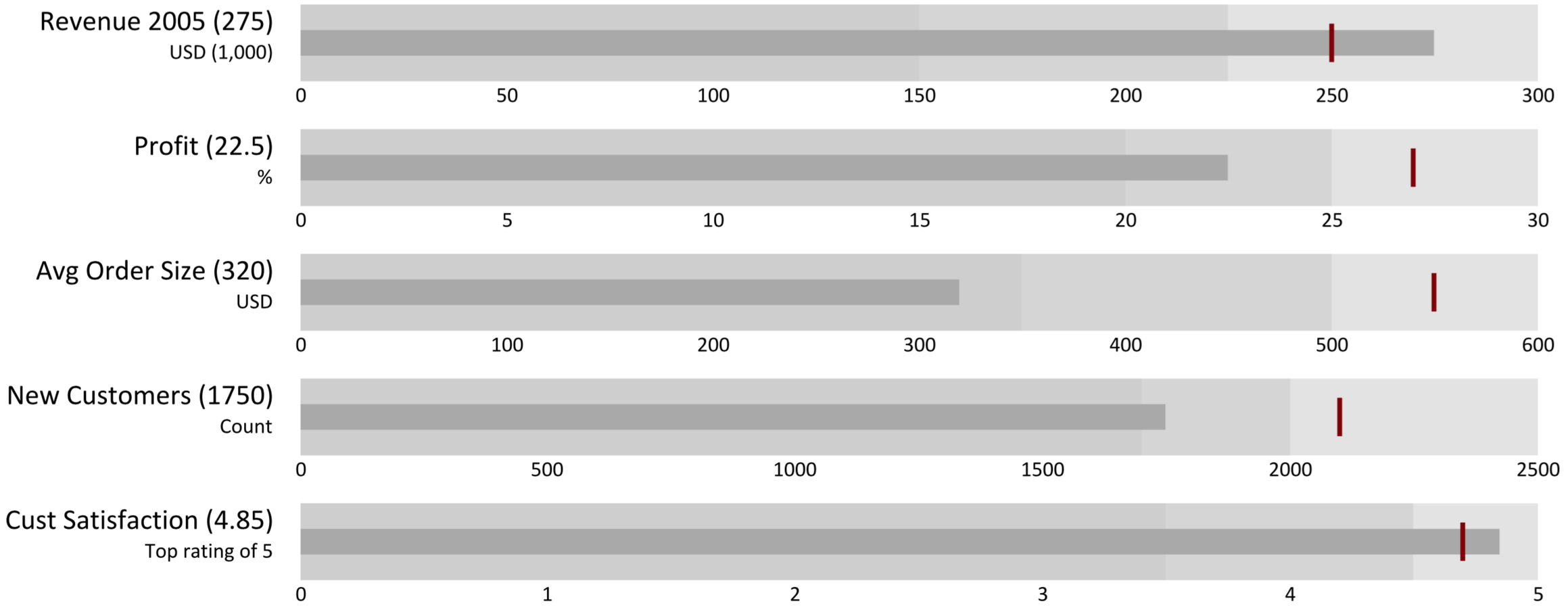




structlayout -json bytes Reader | structlayout-svg -t bytes.Reader > reader.svg

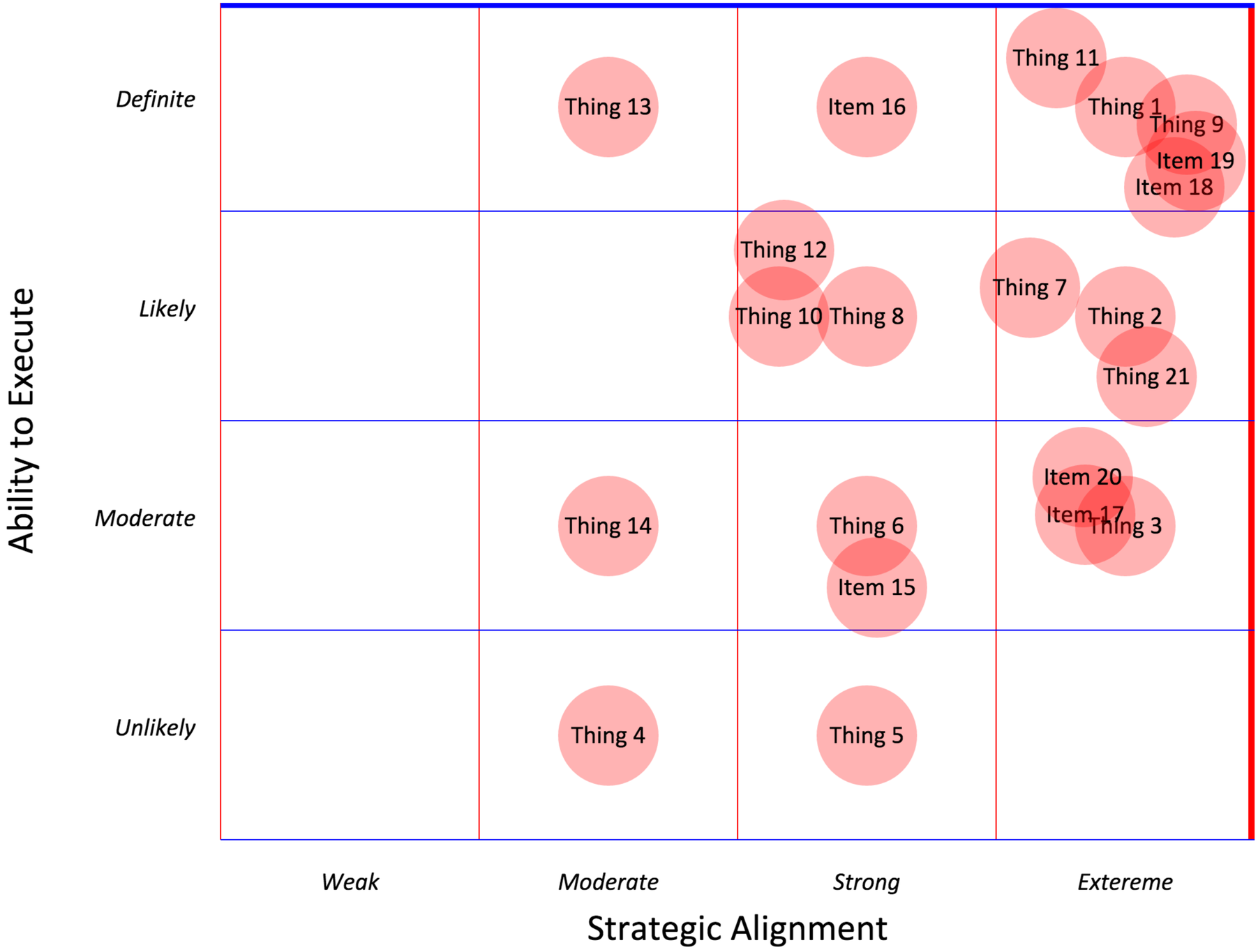
Go Programming Language Release History



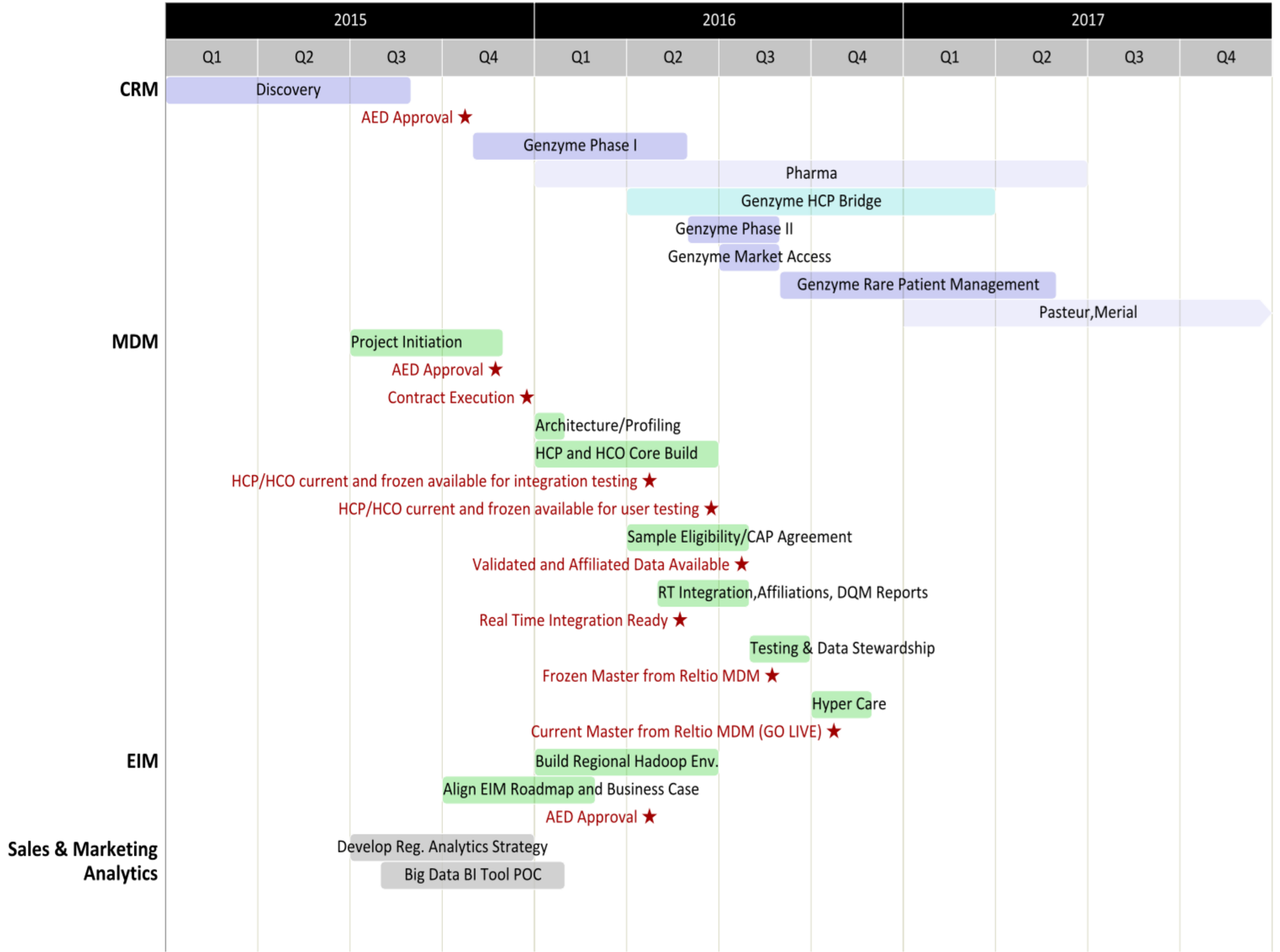


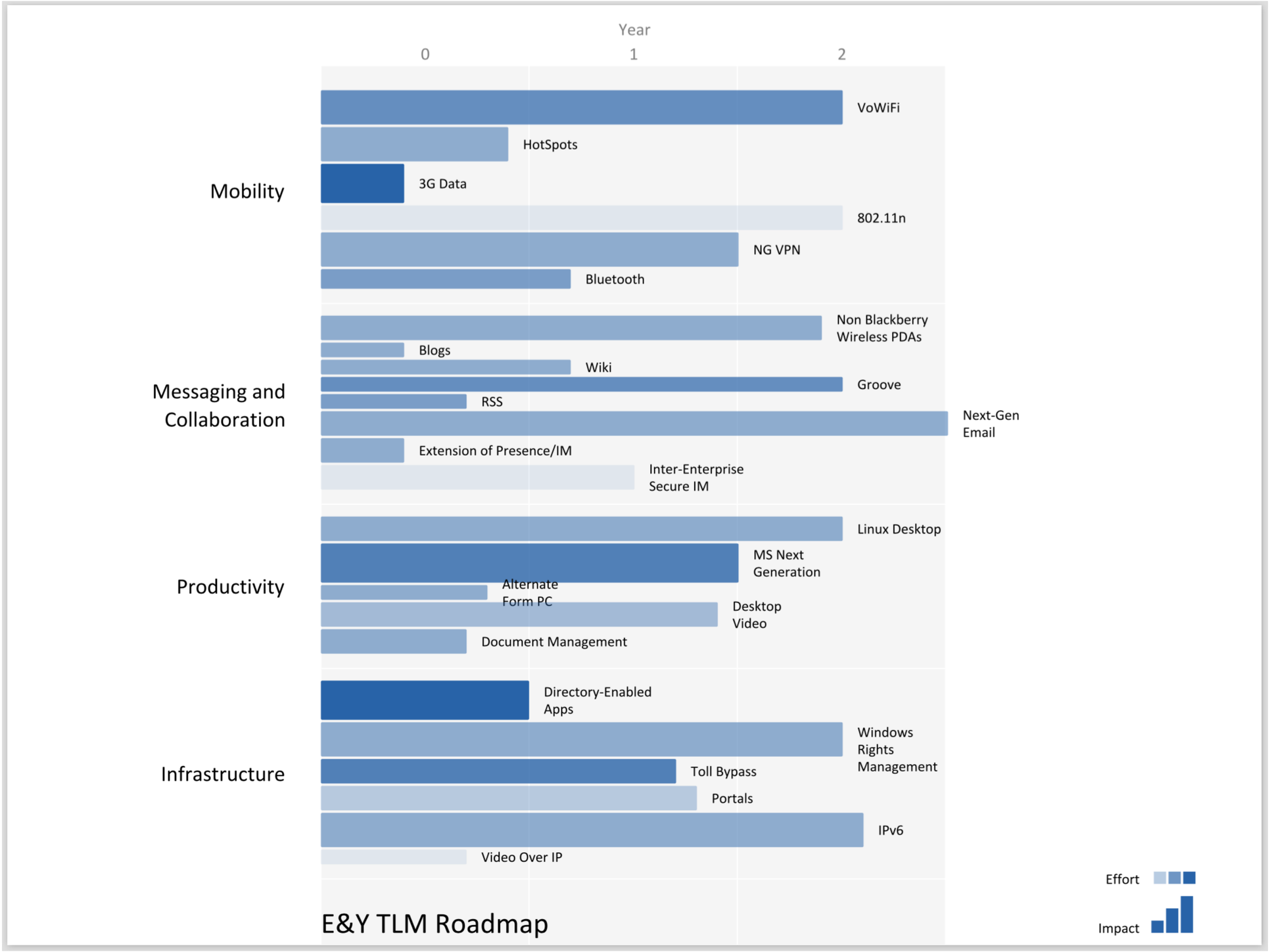
Sample Bullet Graph

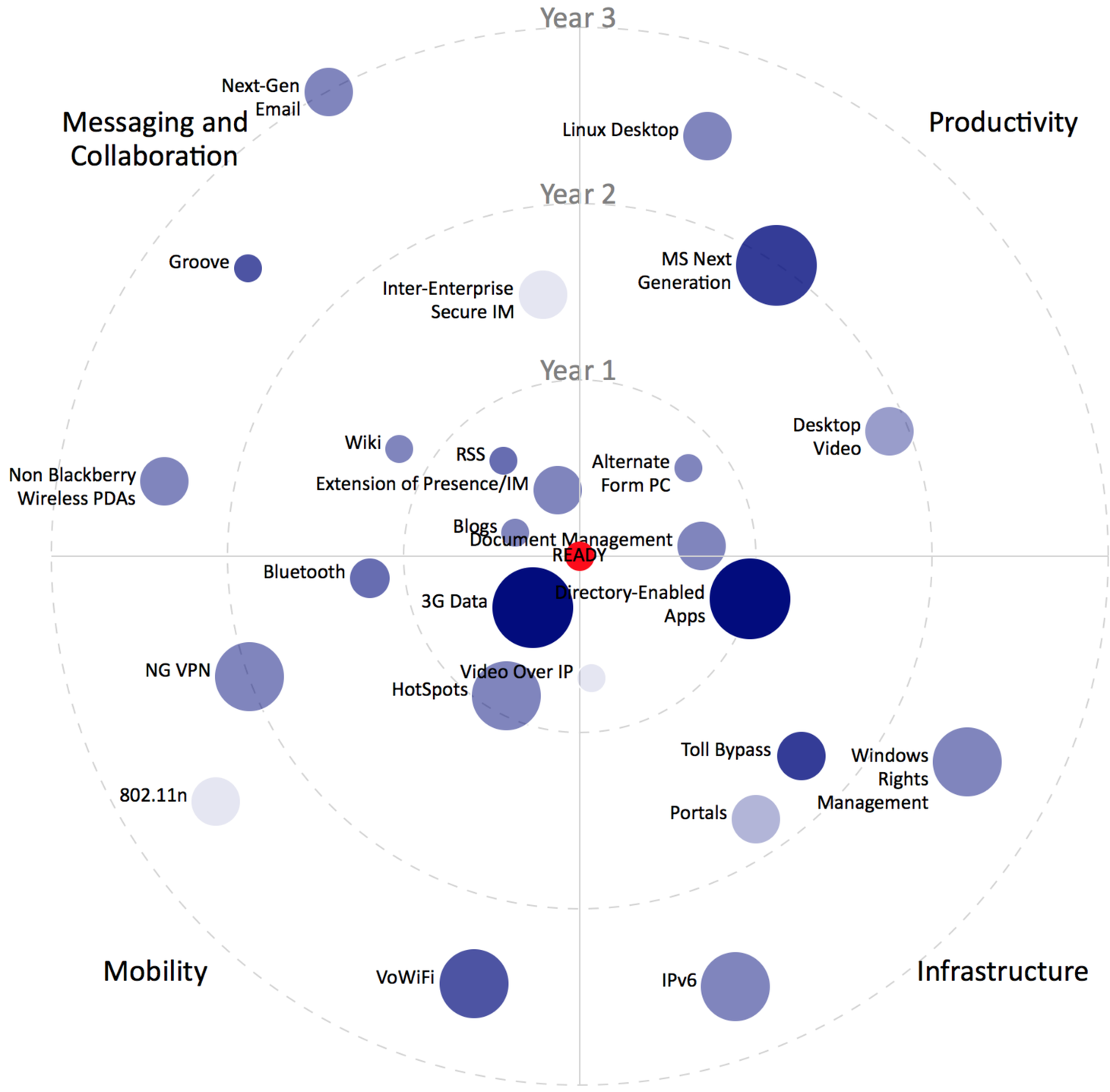
The bullet graph features a single, primary measure (for example, current year-to-date revenue) compares that measure to one or more other measures to enrich its meaning, for example, compared to a target), and displays it in the context of qualitative ranges of performance, such as poor, satisfactory, and good.


Ability to Execute
Definite
Likely
Moderate
Unlikely
Thing 13
Item 16
Thing 11
Thing 1
Thing 9
Item 19
Item 18
Thing 12
Thing 10
Thing 8
Thing 7
Thing 2
Thing 21
Thing 14
Thing 6
Item 15
Item 20
Item 17
Thing 3
Thing 4
Thing 5
Weak
Moderate
Strong
Extereme
Strategic Alignment

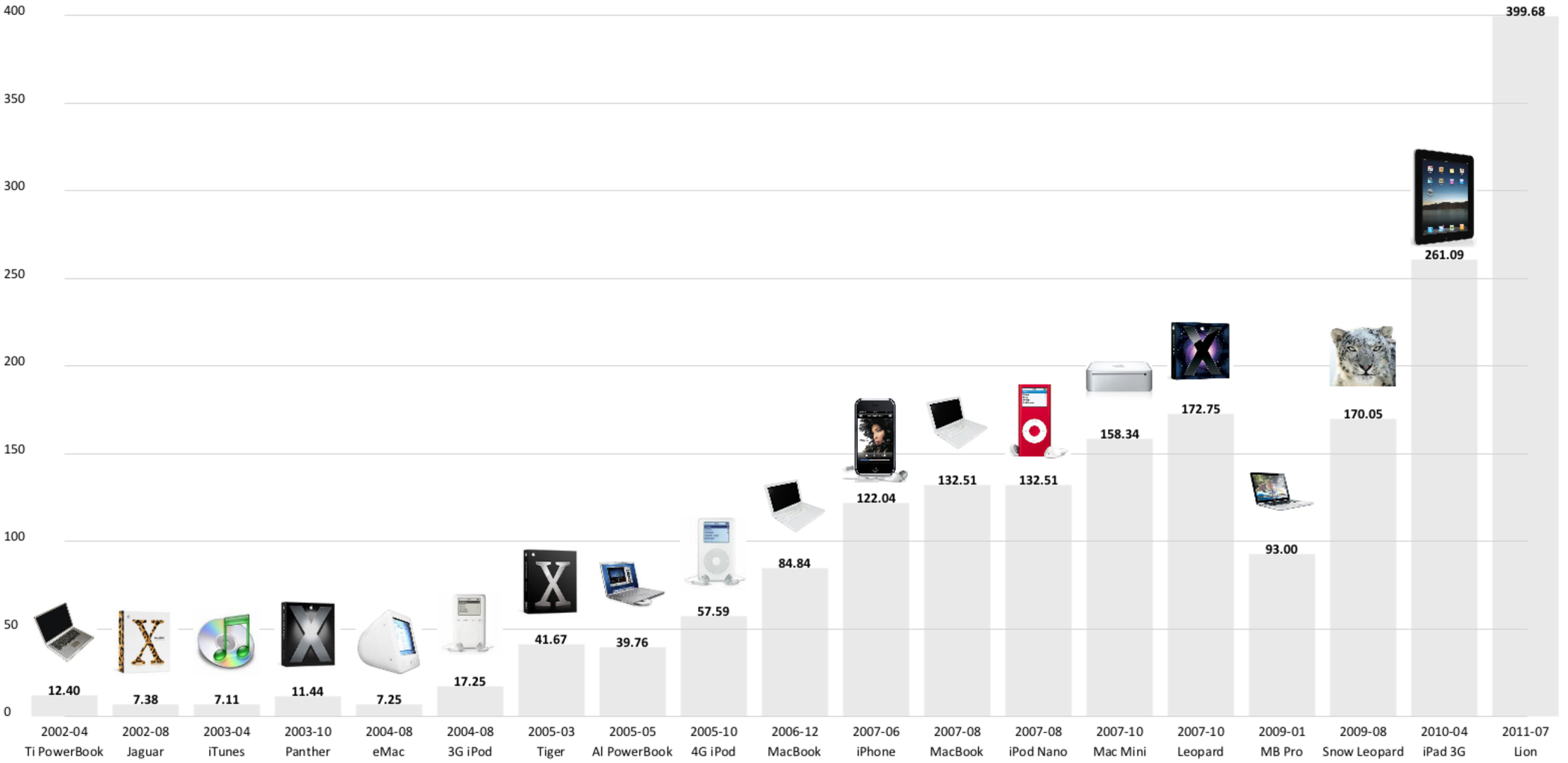
Commercial Capabilities Consolidated Roadmap: 4Q 2015



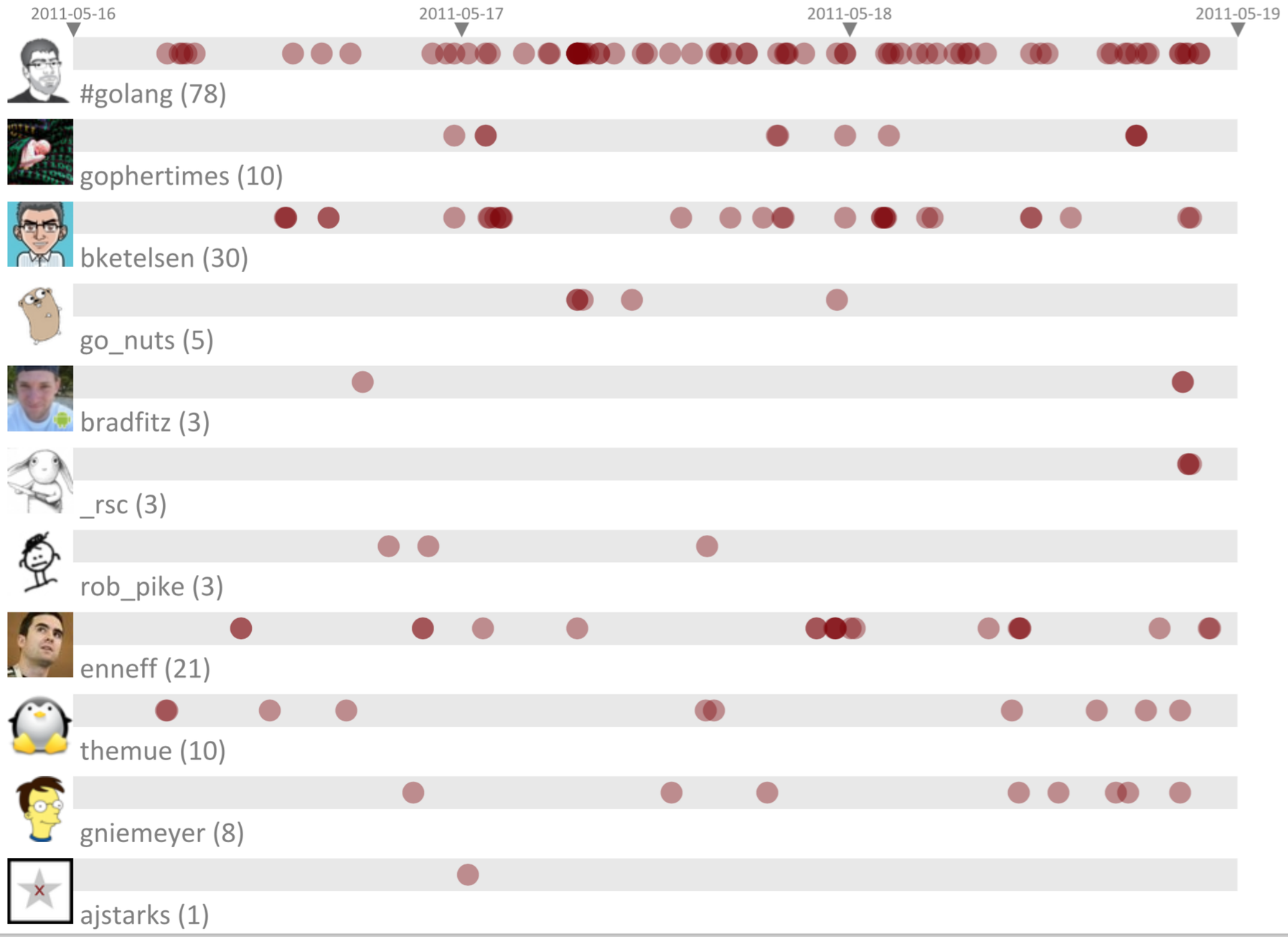

Year
0
1
2
Mobility
VoWiFi
HotSpots
3G Data
802.11n
NG VPN
Bluetooth
Messaging and Collaboration
Non Blackberry Wireless PDAs
Blogs
Wiki
Groove
RSS
Next-Gen Email
Extension of Presence/IM
Inter-Enterprise Secure IM
Productivity
Linux Desktop
MS Next Generation
Alternate Form PC
Desktop Video
Document Management
Infrastructure
Directory-Enabled Apps
Windows Rights Management
Toll Bypass
Portals
IPv6
Video Over IP
E&Y TLM Roadmap
Effort
Impact


Year 3
Year 2
Year 1
Messaging and Collaboration
Productivity
Mobility
Infrastructure
Next-Gen Email
Linux Desktop
Groove
Inter-Enterprise Secure IM
MS Next Generation
Desktop Video
Wiki
RSS
Alternate Form PC
Non Blackberry Wireless PDAs
Extension of Presence/IM
Blogs
Document Management
READY
Bluetooth
3G Data
Directory-Enabled Apps
NG VPN
Video Over IP
HotSpots
Toll Bypass
Windows Rights Management
802.11n
Portals
VoWiFi
IPv6

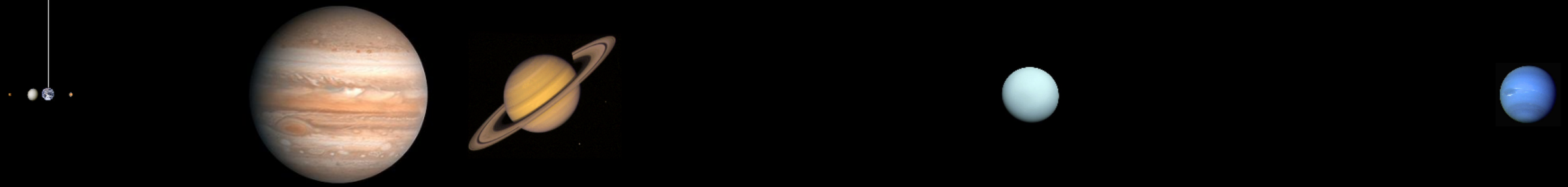
Apple Purchases and Stock Price



Twitter Update Frequency



You are here

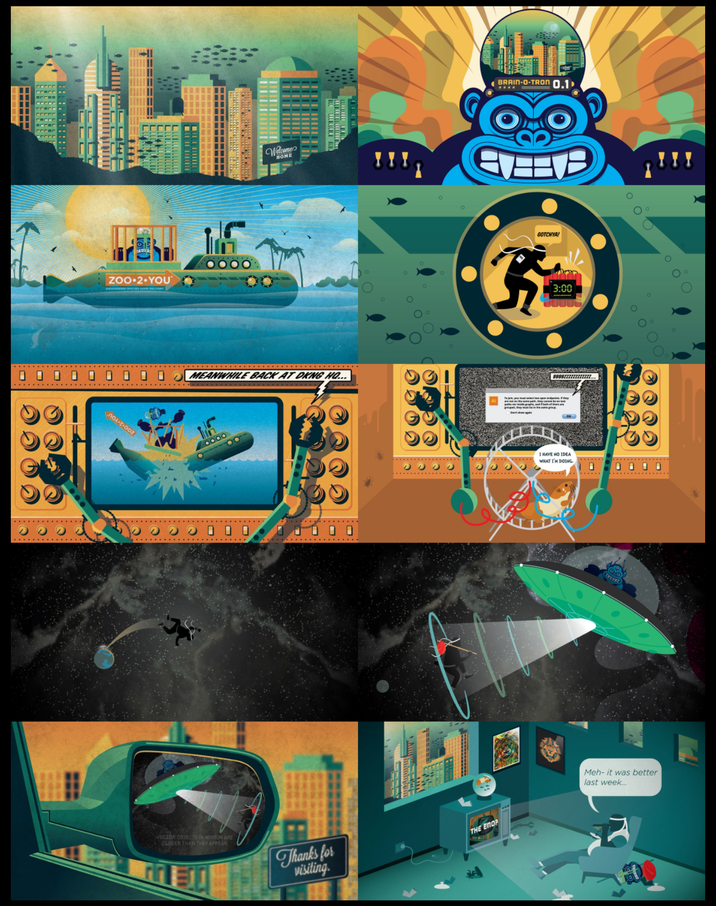


JONES
MONTIEL

- LAYER TENNIS SEASON 4 WEEK 6 MATCH COMMENTARY BY MIKE MONTEIRO -

DKNG
DDL

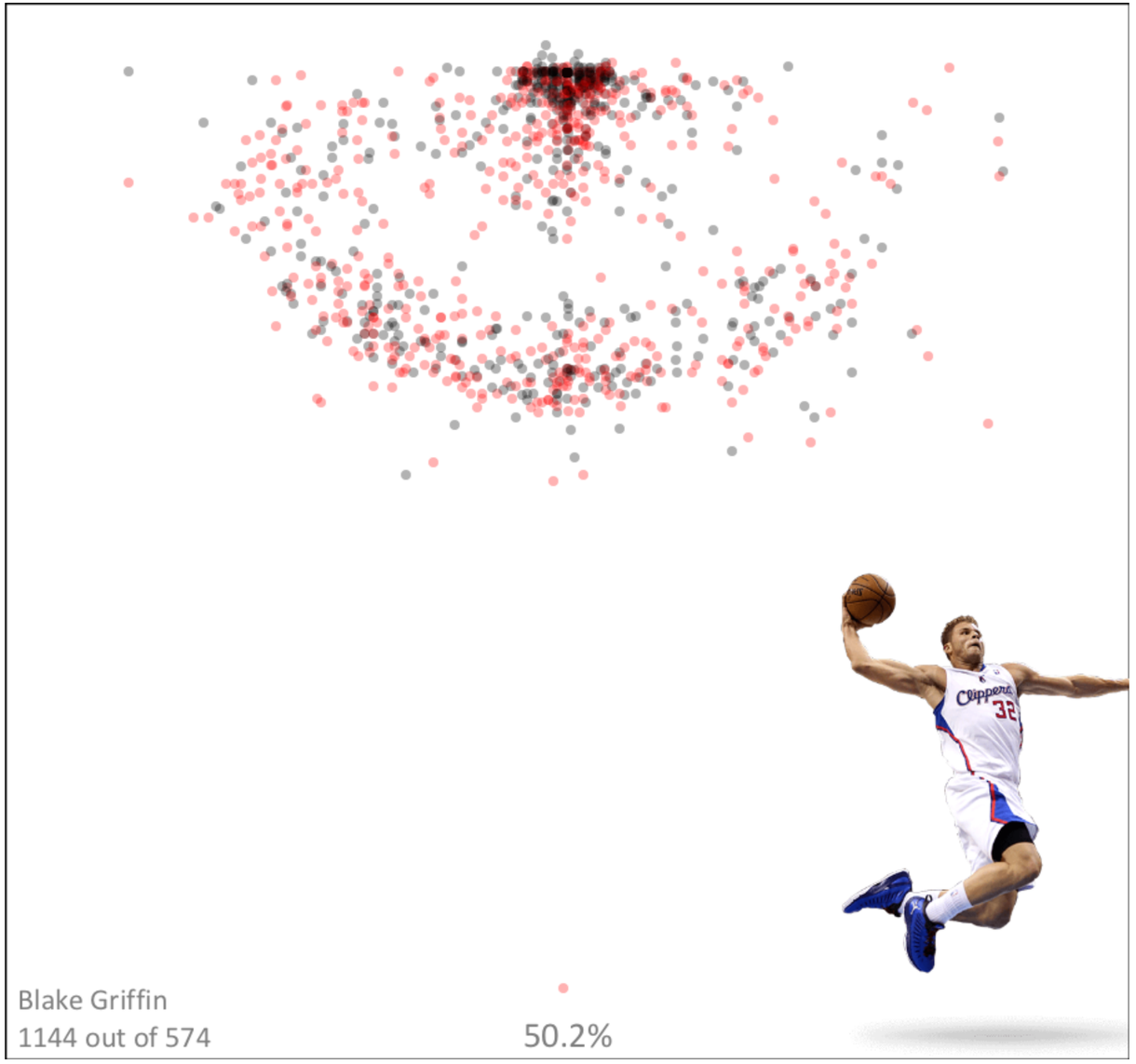
- LAYER TENNIS SEASON 4 WEEK 7 MATCH COMMENTARY BY JOHN GRUBER -

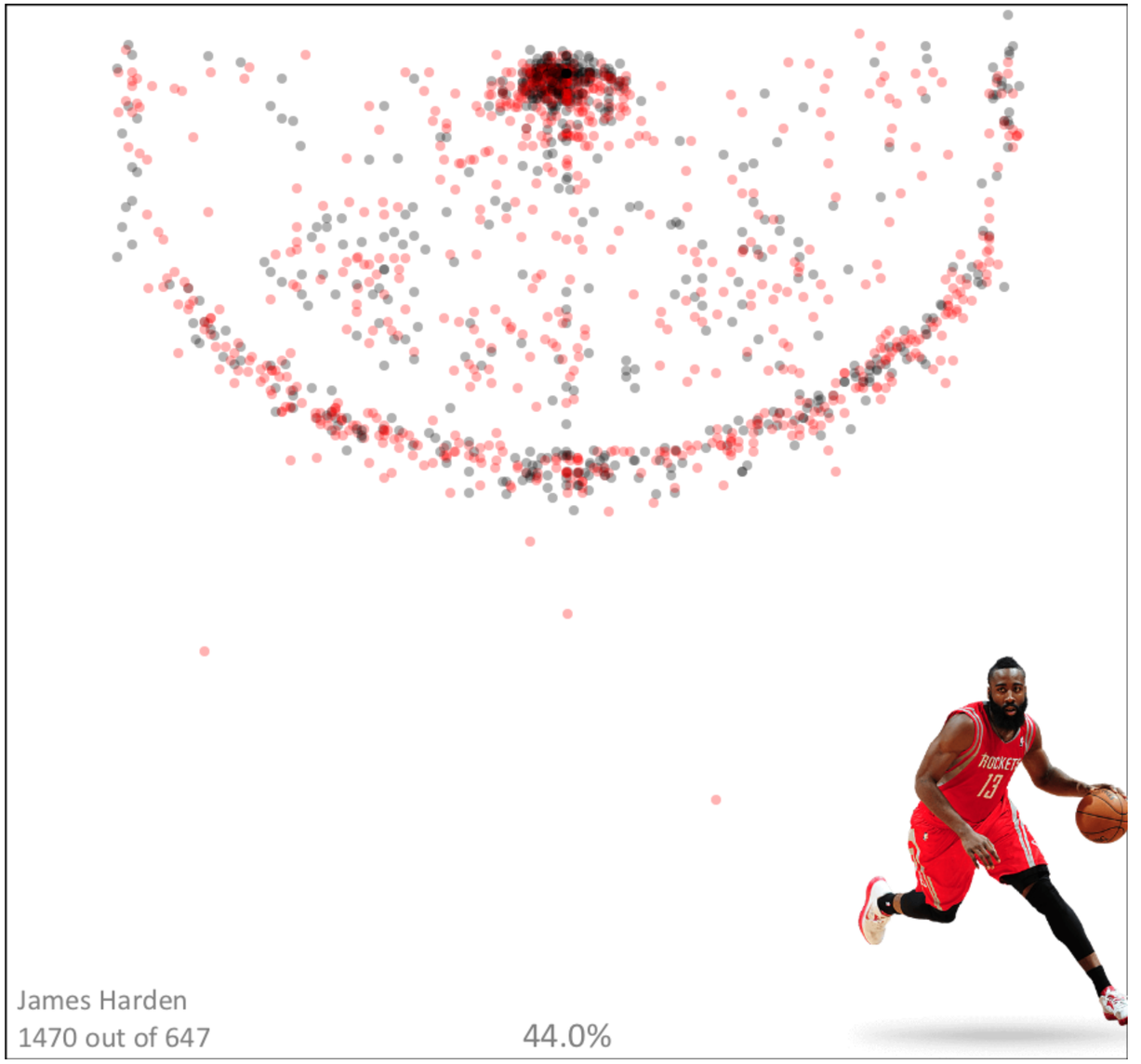


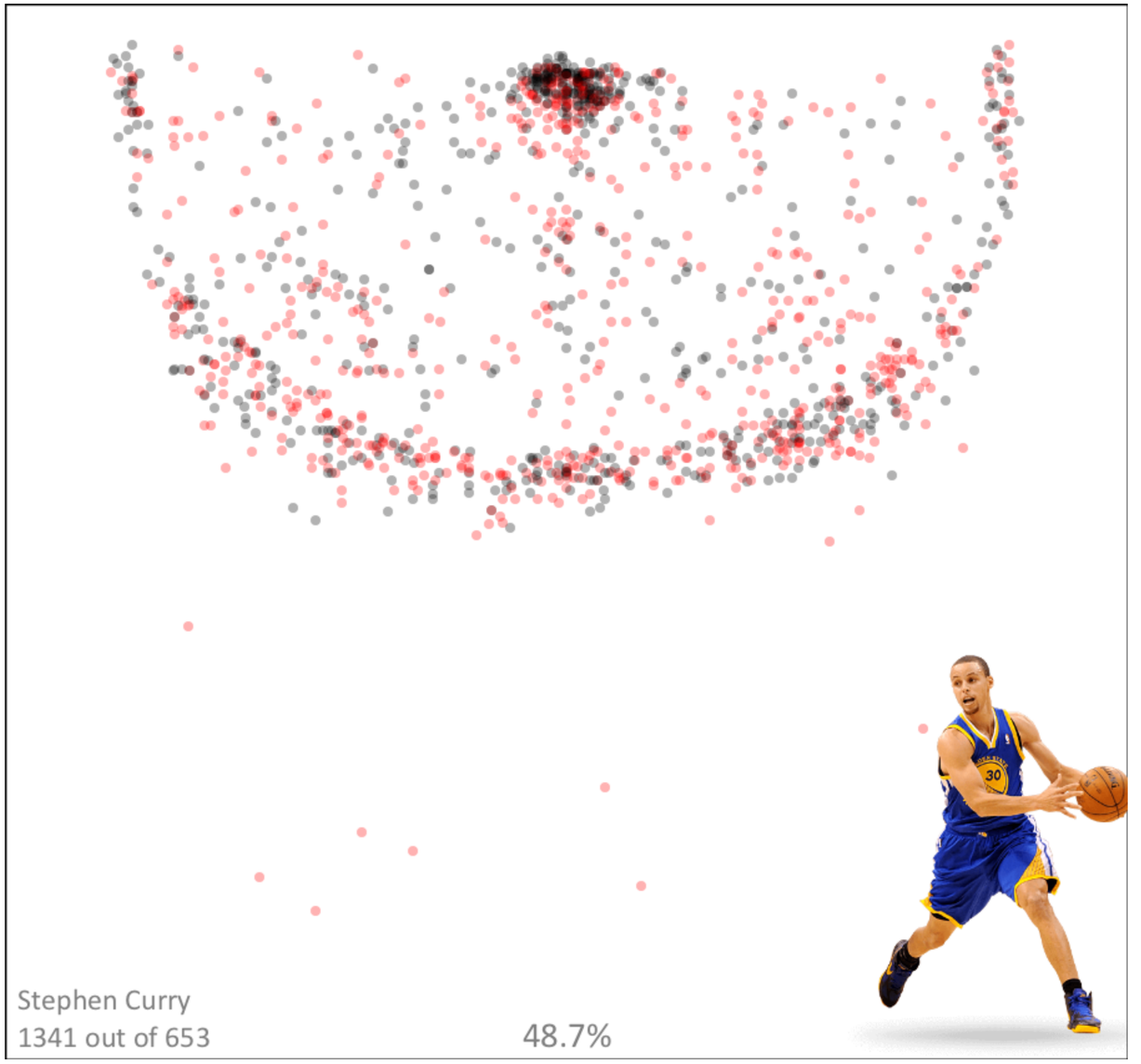
ANDERSON
DKNG

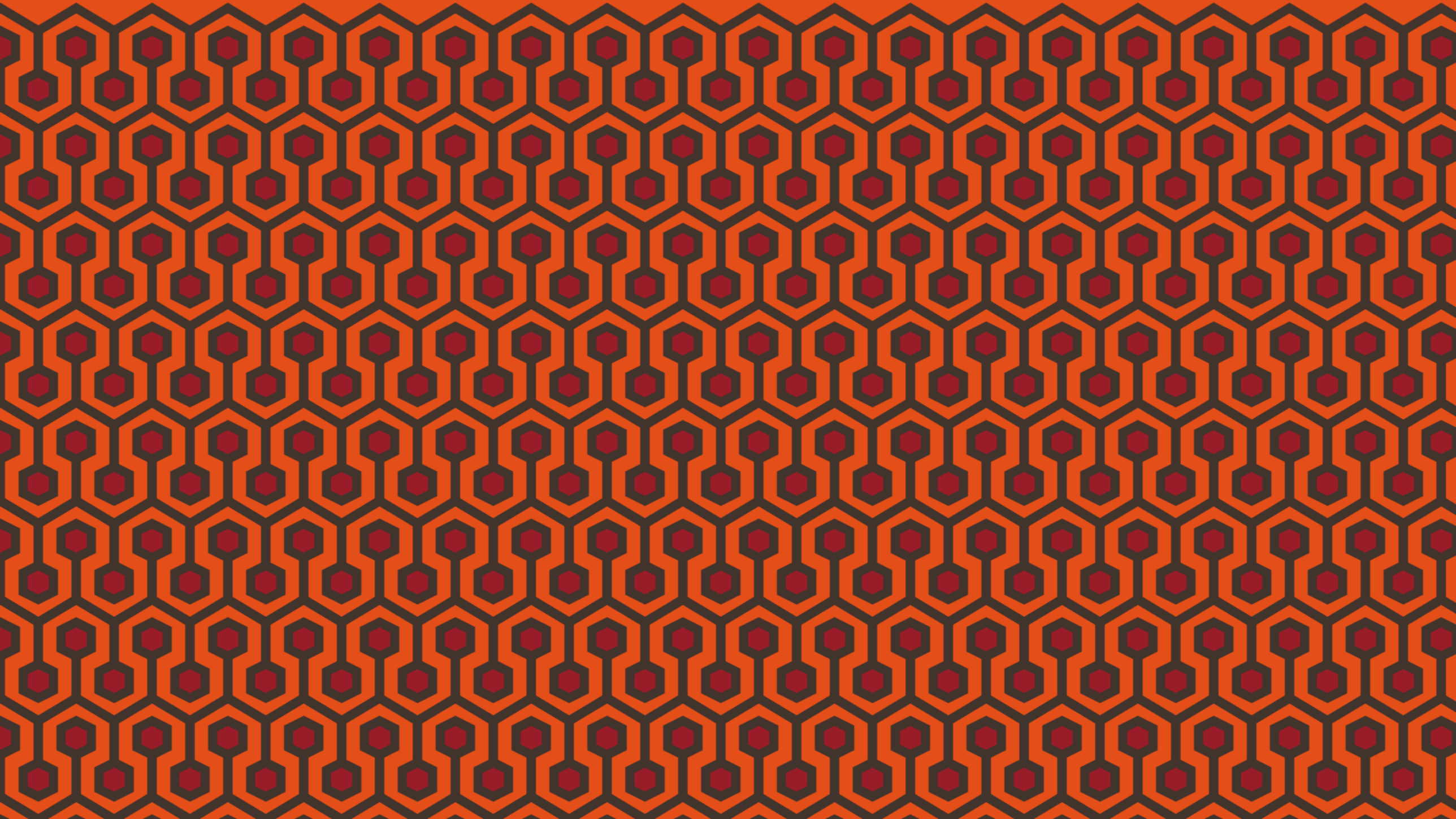
- LAYER TENNIS SEASON 4 PLAYOFF QUARTERFINAL COMMENTARY BY BRYAN BEDELL -


CONTINO CASSARO
PUTNAM STOCKS
WHITE TAYLOR
HALL WARREN
STRAWBERRYLUNA DOUBLENAUT
JONES MONTIEL
DKNG DDL
SHAWNA X STEVENS
TAYLOR JONES
PUTNAM RAJKUMAR
REYES WHITE
ANDERSON DKNG
TAYLOR WHITE
RAJKUMAR ANDERSON
ANDERSON WHITE
A FIGHT TO THE DEATH!
THE BATTLE OF LONG ISLAND
DIVIDE AND CONQUER
DIVIDE AND CONQUER?
JON CONTINO SOMEHOW TURNS INTO AN EAGLE!!
CUT OFFS
2022: A SPACE ODDITY
LANDING... 5 4 3 2 1
The Great British Autumn
Autumn HARVEST
OR
THEY MAY EAT YOUR PENGUIN
WHAT THE PUCK?
DECK THE HALL THIS CHRISTMAS EVE
HALLO ON A SEC AREN'T WE GETTING A BIT HASTEVE?! IT'S ONLY OCTOBER
SORRY AM I THE ONLY ONE WITH A TIME MACHINE?
WE SAY ABOUT THEY SAY A
REGULAR
SUPER SIZE ME!
VISIT THE "GREAT" WHITE NORTH
THE GREAT WHITE NORTH
CANADIANS HAVE BIGGER BALLS
"OUR BEER IS BETTER" QUOTE GOES HERE.
EVERYONE KNOWS THAT'S NOT TRUE
bronze
bro hoe
whamf
PENGUIN BURGER
It wasn't me
She won't be missed for long.
CASSANDRA THUNDER
07:30
OINK
OOPS!
SNOOZE
200M
LET'S GO CRAZY


Blake Griffin
1144 out of 574
50.2%

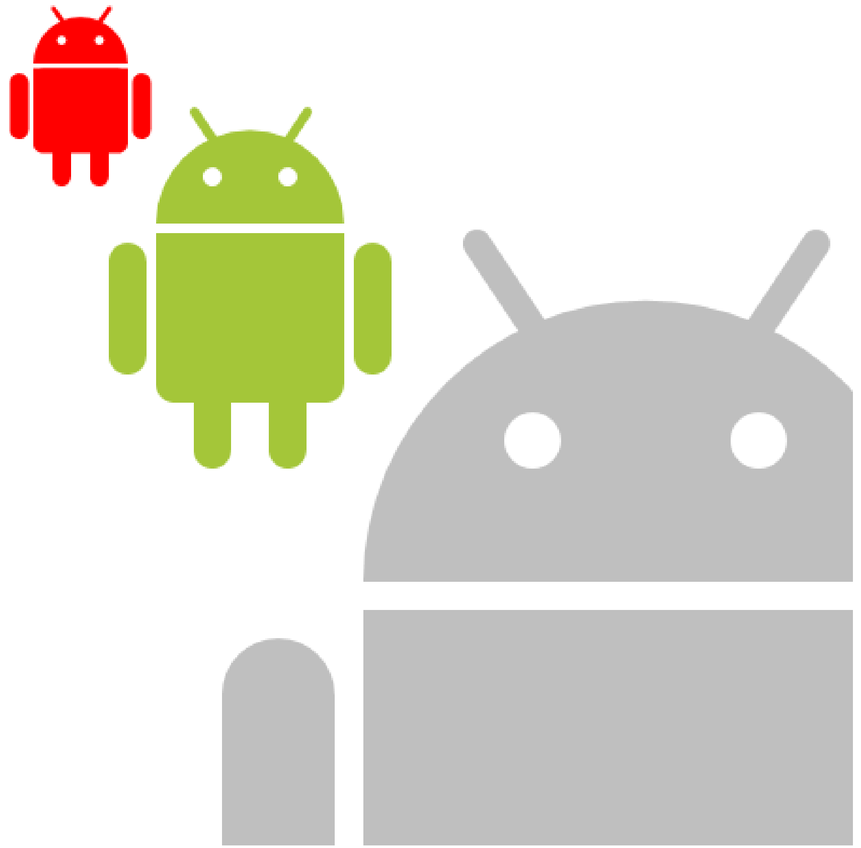

James Harden
1470 out of 647
44.0%

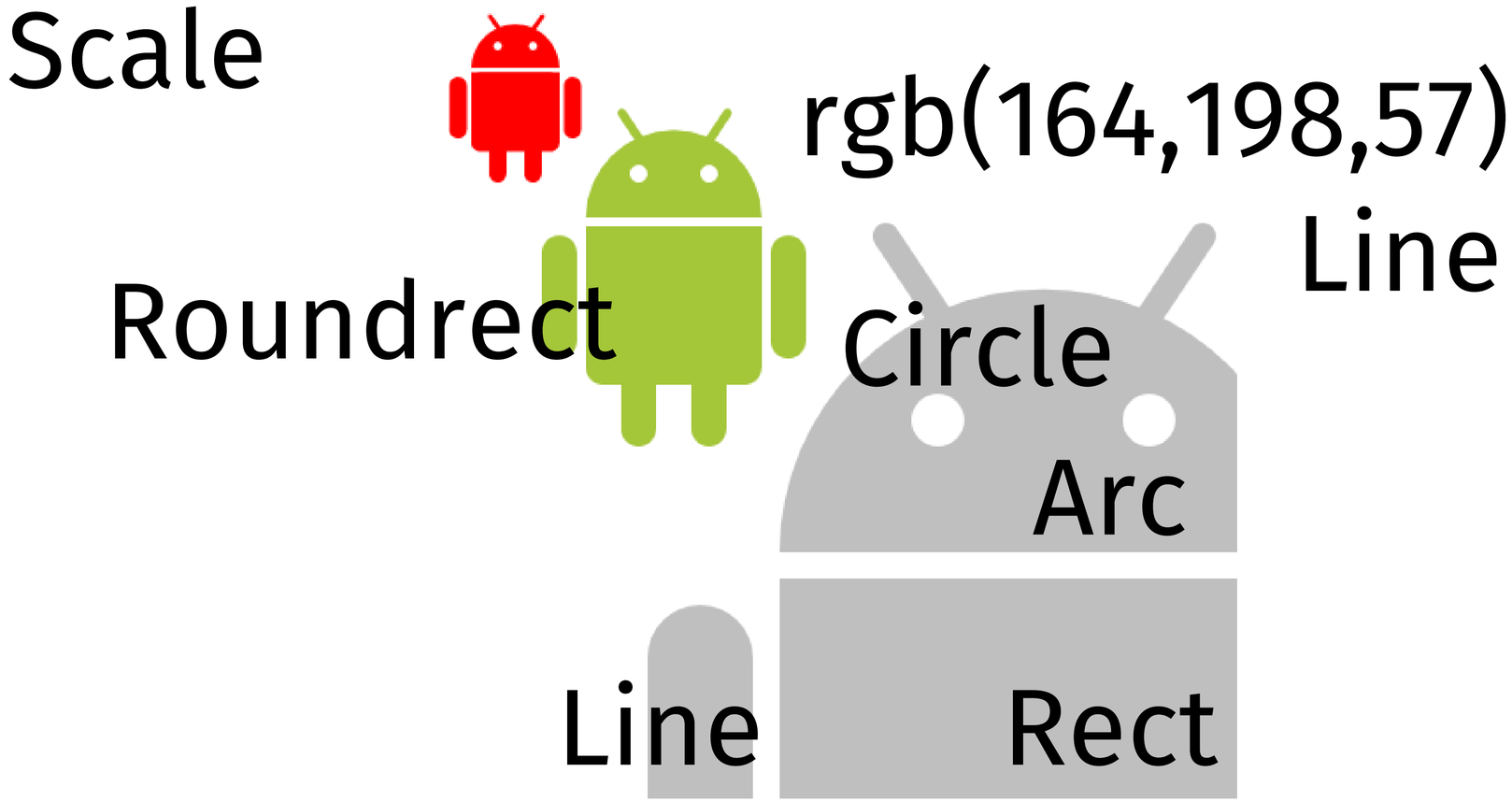

Stephen Curry
1341 out of 653
48.7%



```
canvas.Def()
canvas.Gid("unit")
canvas.Polyline(xl, yl, "fill:none")
canvas.Polygon(xp, yp)
canvas.Gend()

canvas.Gid("runit")
canvas.TranslateRotate(150, 180, 180)
canvas.Use(0, 0, "#unit")
canvas.Gend()
canvas.Gend()
canvas.DefEnd()

for y := 0; < height; y += 130 {
    for x := 100; x < width; x+=100 {
        canvas.Use(x, y, "#unit")
        canvas.Use(x, y, "#runit")
    }
}
```

SVGo API Design

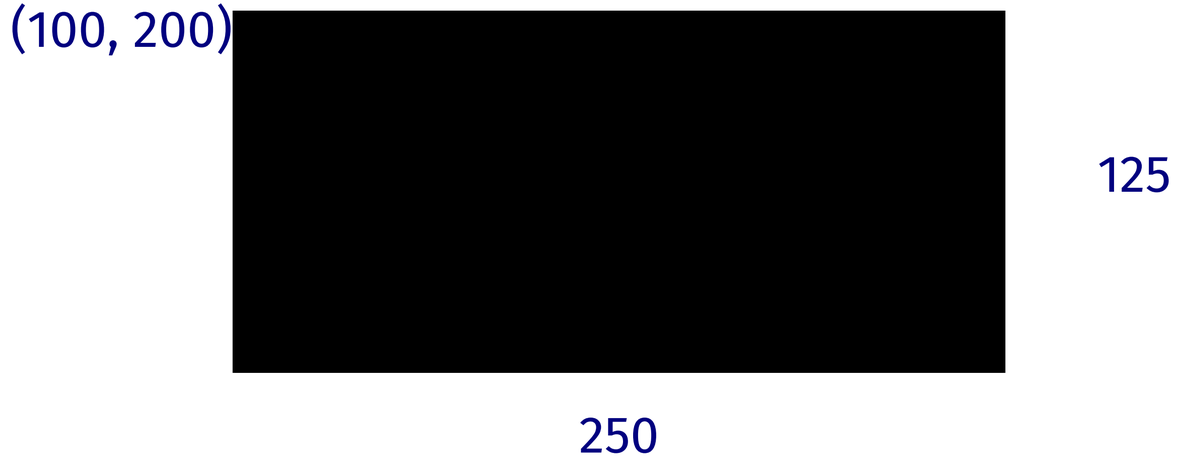



Scale
rgb(164,198,57)
Line
Roundrect
Circle
Arc
Line
Rect

Element | Arguments

Rect (100,200,250,125)

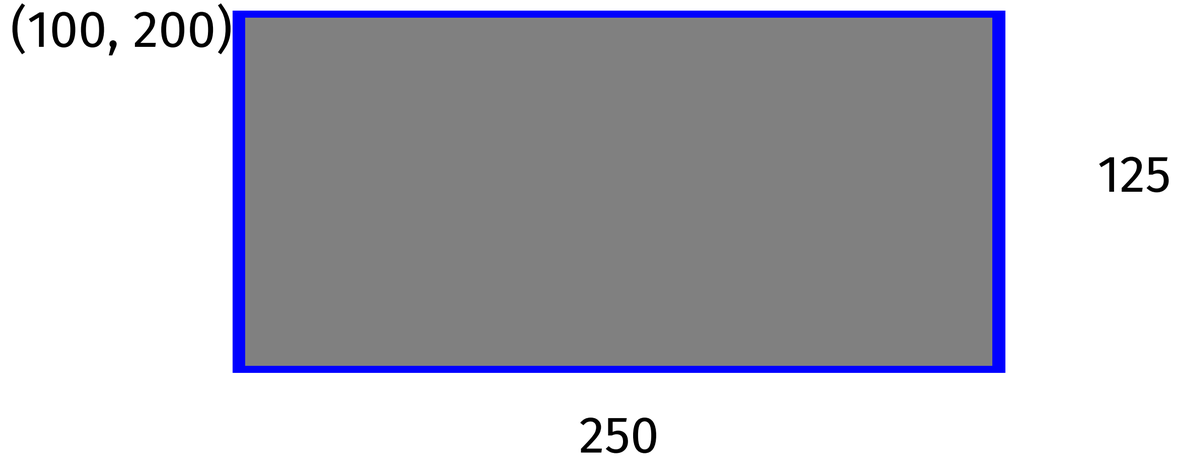
```
<rect x="100" y="200" width="250" height="125"/>
```



Element Arguments CSS Style

```
Rect (100,200,250,125,
      "fill:gray;stroke:blue")
```

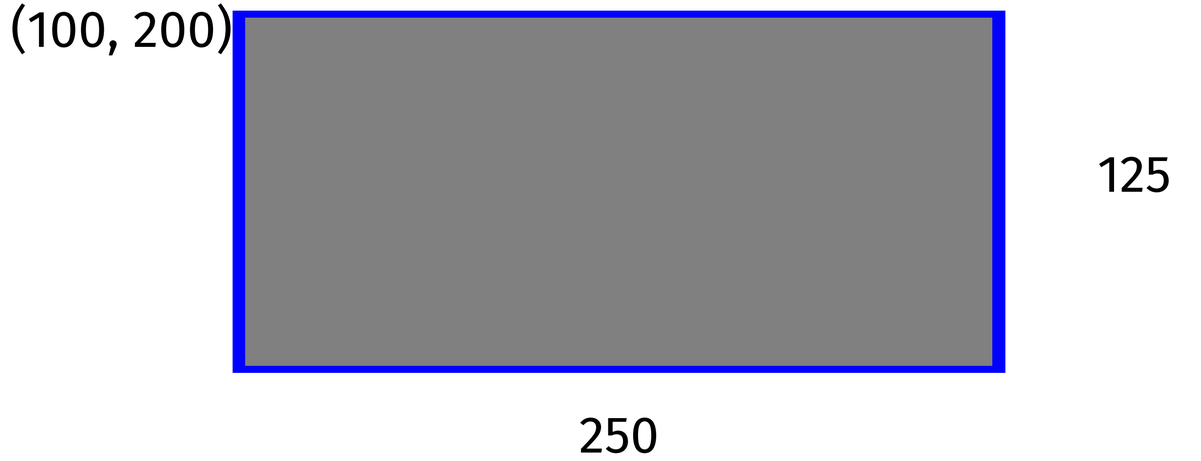
```
<rect x="100" y="200" width="250" height="125"
      style="fill:gray;stroke:blue"/>
```

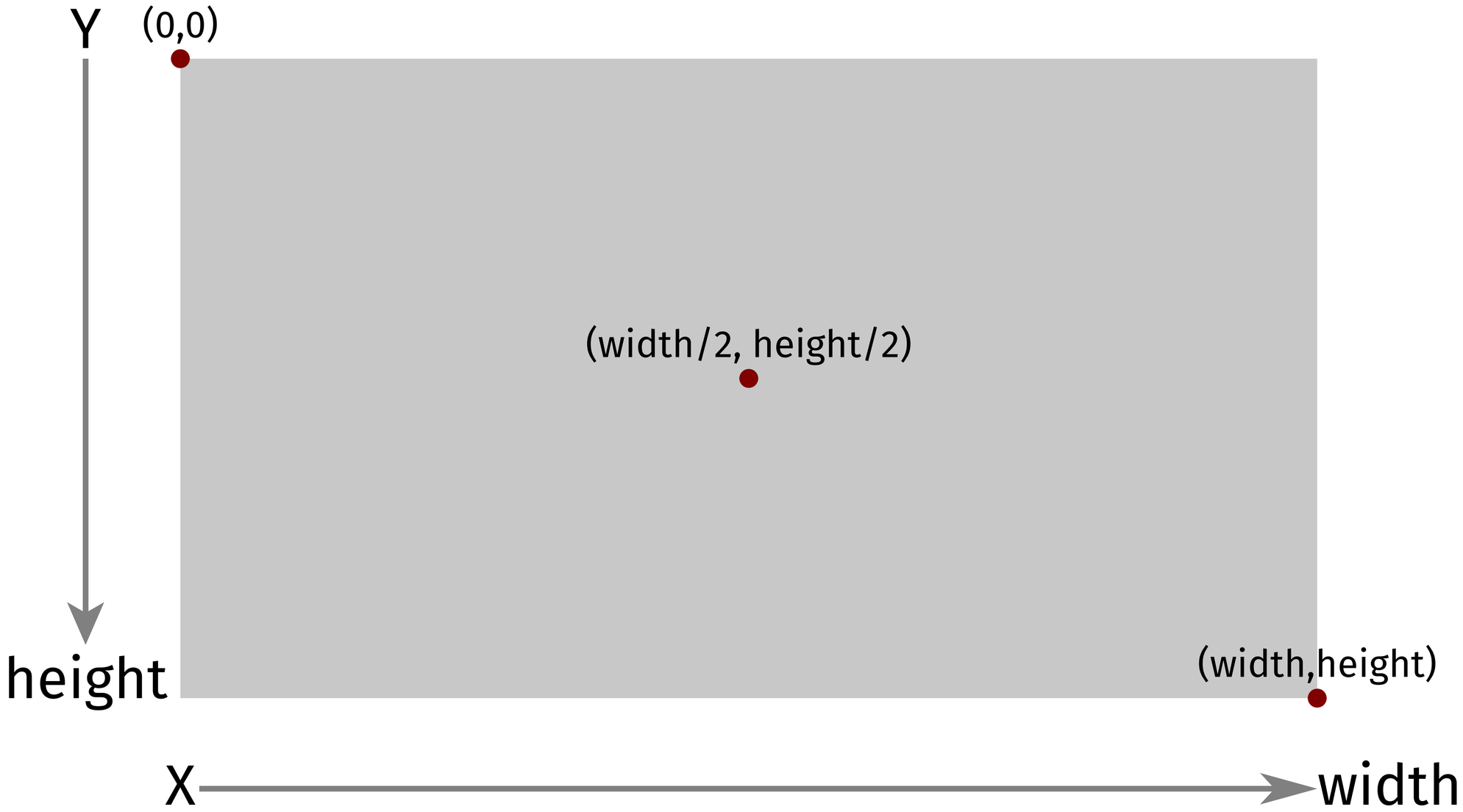


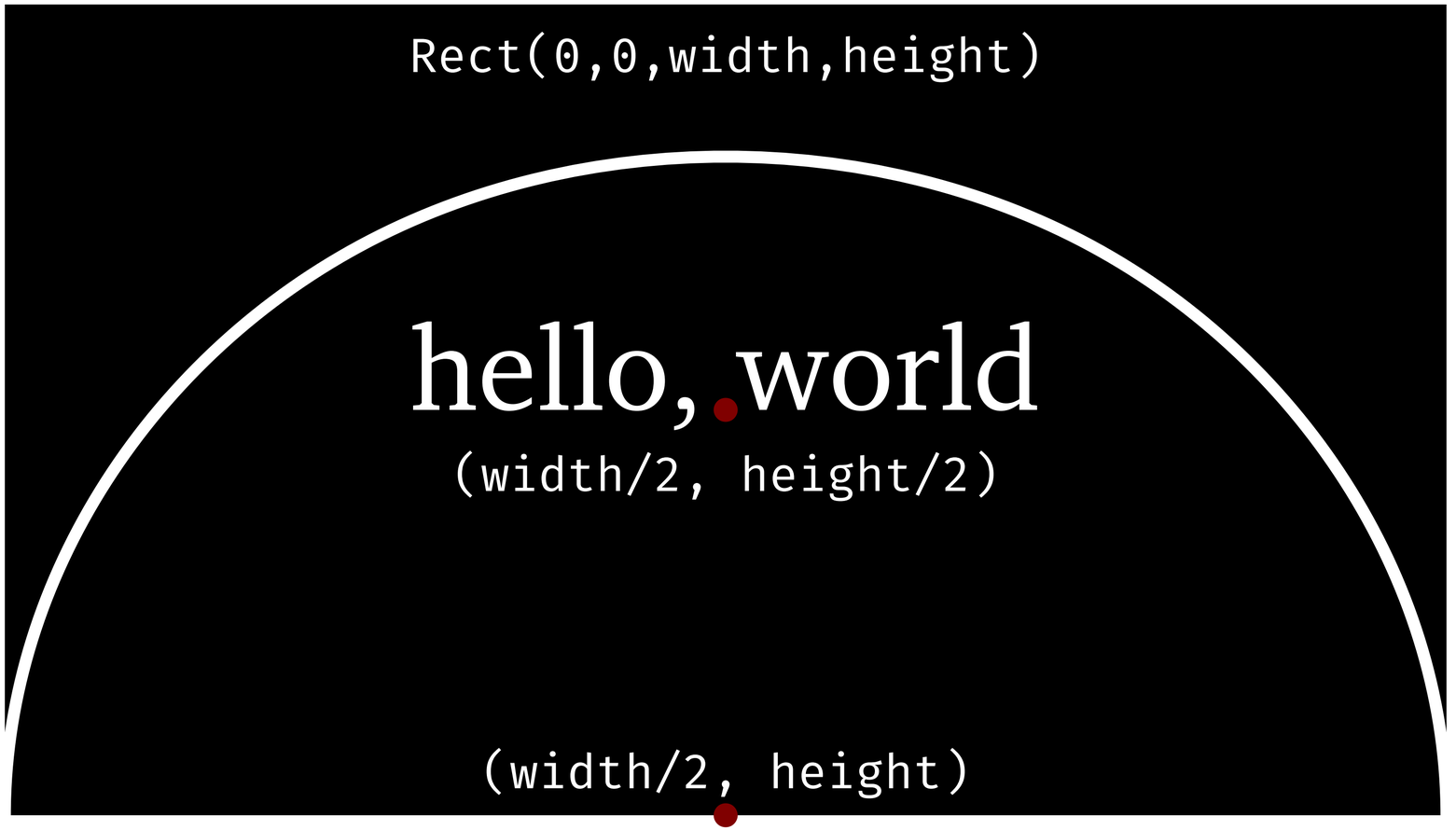
Element Arguments Attributes

```
Rect (100,200,250,125,
    `id='box'`, `fill='gray'`, `stroke='blue'`)
```

```
<rect x="100" y="200" width="250" height="125"
      id='box' fill='gray' stroke='blue'/>
```




Y
(0,0)
(width/2, height/2)
height
(width,height)
X
width

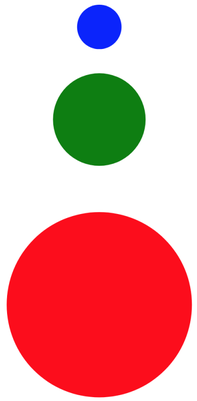

Rect(0,0,width,height)
hello, world
(width/2, height/2)
(width/2, height)

```
package main

import (
    "os"
    "github.com/ajstarks/svgo"
)

func main() {
    width := 960
    height := 540
    style := "fill:white;font-size:60pt;text-anchor:middle"
    canvas := svg.New(os.Stdout)
    canvas.Start(width, height)
    canvas.Rect(0, 0, width, height)
    canvas.Circle(width/2, height, width/2, "fill:rgb(44,77,232)")
    canvas.Text(width/2, height/2, "hello, world", style)
    canvas.End()
}
```

Data

```
<thing top="100" left="100" sep="100">
    <item width="50"  height="50"  name="Little" color="blue">This is small</item>
    <item width="75"  height="100" name="Med"    color="green">This is medium</item>
    <item width="100" height="200" name="Big"    color="red">This is large</item>
</thing>
```


Picture

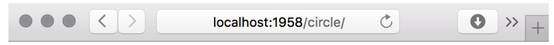


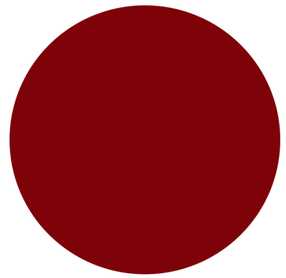
Little:This is small/blue

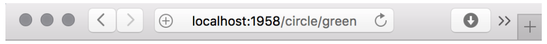
Med:This is medium/green

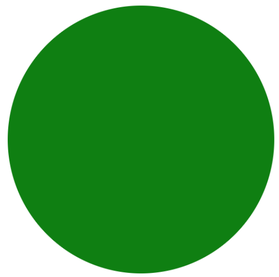
Big:This is large/red

```
const defaultstyle = "fill:rgb(127,0,0)"

func circle(w http.ResponseWriter, req *http.Request) {
    w.Header().Set("Content-Type", "image/svg+xml")
    s := svg.New(w)
    s.Start(500, 500)
    s.Title("Circle")
    s.Circle(250, 250, 125, shapestyle(req.URL.Path))
    s.End()
}

func shapestyle(path string) string {
    i := strings.LastIndex(path, "/") + 1
    if i > 0 && len(path[i:]) > 0 {
        return "fill:" + path[i:]
    }
    return defaultstyle
}
```



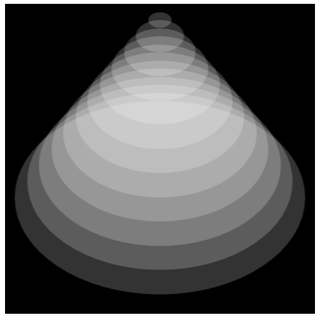




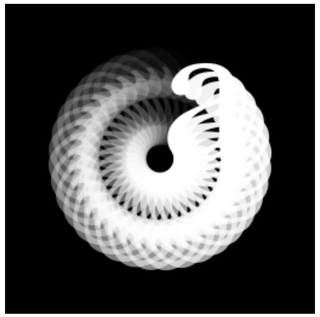


https://ajstarks.org:1958/

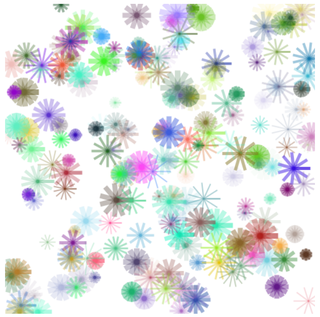
clock



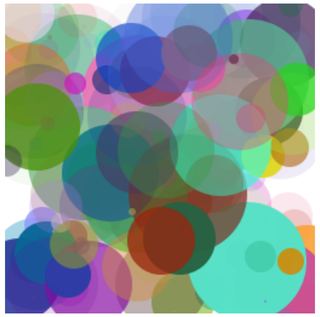
funnel



rotext

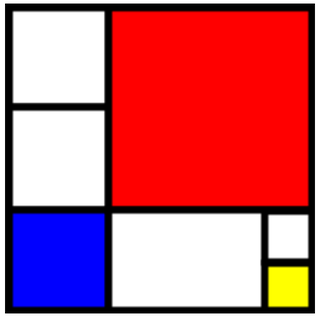


flower

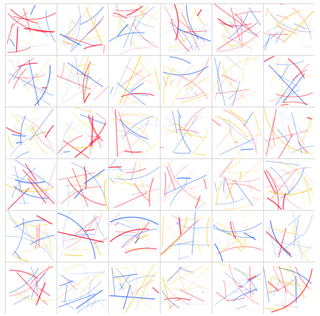


rshape

cube



mondrian

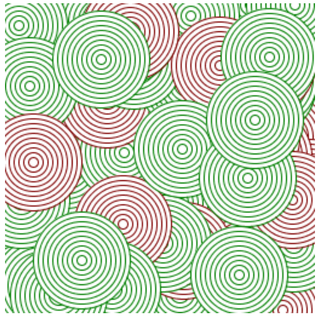


lewitt

face

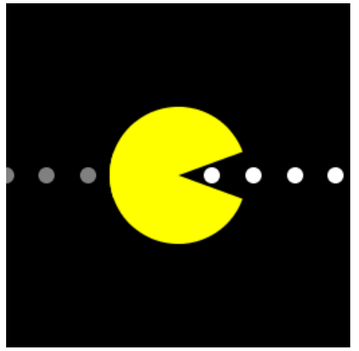
pacman

tux



concentric

/pacman/?angle=20



/pacman/?angle=40

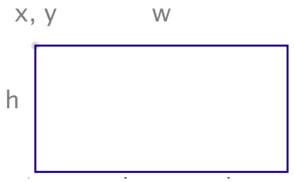
/pacman/?angle=70

API Reference

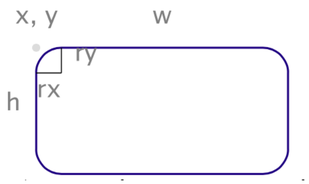
SVGo Method Categories

- Starting and Ending
- Metadata
- Utility
- Groups and Definitions
- Markers and Patterns
- Links
- Masking and Clipping
- Transforms
- Skewing
- Shapes
- Paths, Lines and Curves
- Images and Text
- Colors and Gradients
- Filter Effects

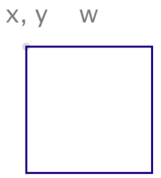
Shapes



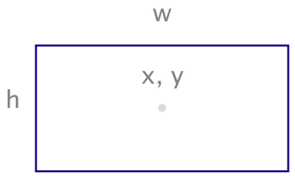
Rect(x, y, w, h int, s string...)



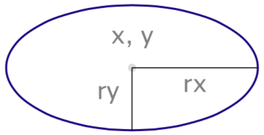
Roundrect(x, y, w, h, rx, ry int, s string...)



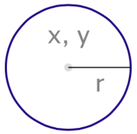
Square(x, y, l int, s string...)



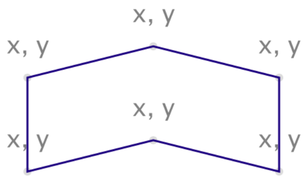
CenterRect(x, y, w, h int, s string...)



Ellipse(x, y, w, h int, s string...)

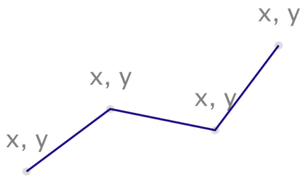


Circle(x, y, r int, s string...)



Polygon(x, y []int, s string...)

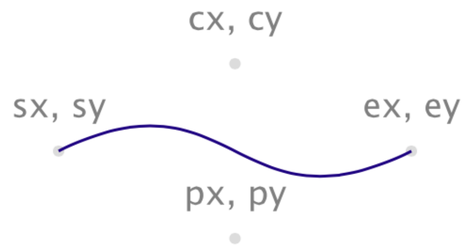
Line(x1, y2, x2, y2 int, s string...)



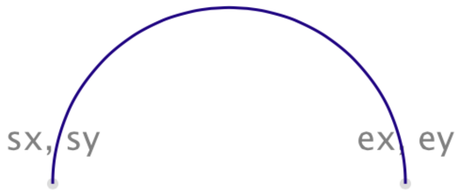
Polyline(x, y []int, s string...)

Curves and Paths

Bezier(sx, sy, cx, cy, ex, ey int, s string...)



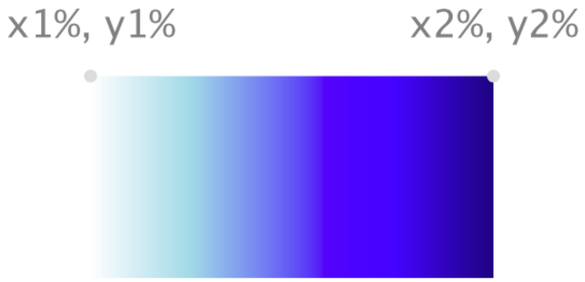
QBez(sx, sy, cx, cy, px, py, ex, ey int, s string...)



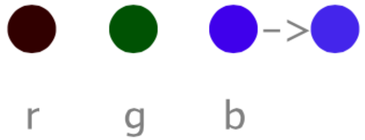
Arc(sx, ax, ay, r int, dir, large bool, ex, ey int, s string...)

Path(d string, s string...)

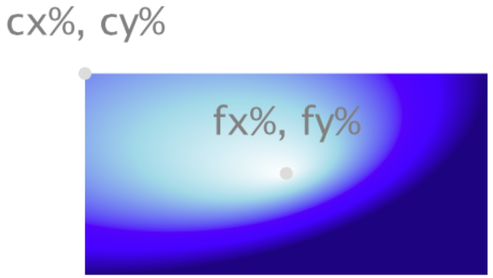
Gradients and Colors



LinearGradient(id string, x1, y1, x2, y2 uint8, sc []Offcolor)



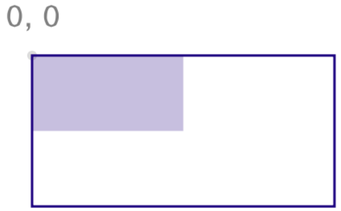
RGB(r, g, b int)



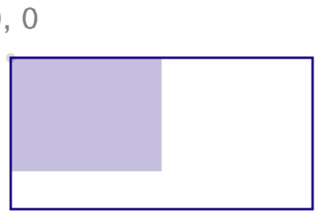
RadialGradient(id string, cx, cy, r, fx, fy uint8, sc []Offcolor)

RGBA(r, g, b int, a float64)

Transforms

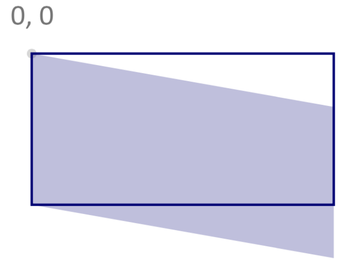


Scale(n float64)

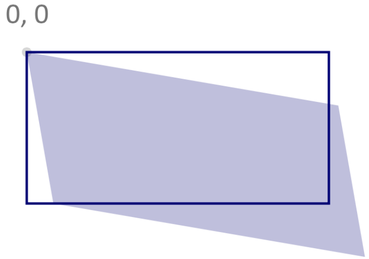


ScaleXY(dx, dy float64)

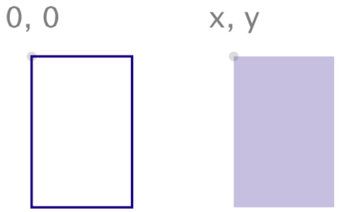
SkewX(a float64)



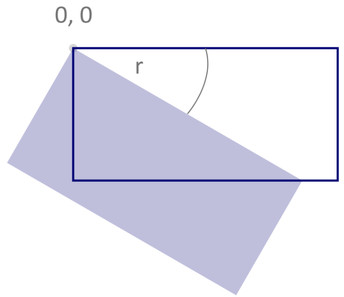
SkewY(a float64)



SkewXY(ax, ay float64)



Translate(x, y int)



Rotate(a float64)

Text and Images

hello, this is SVG

x, y

Text(x, y int, s string, s ...string)

Textpath(t string, pathid string, s ...string)

New(w io Writer)
Start(w, h int, options ...string)/End()
Startview(w, h, minx, miny, vw, vh int)
Group(s ...string)/Gend()
Gstyle(s string)/Gend()

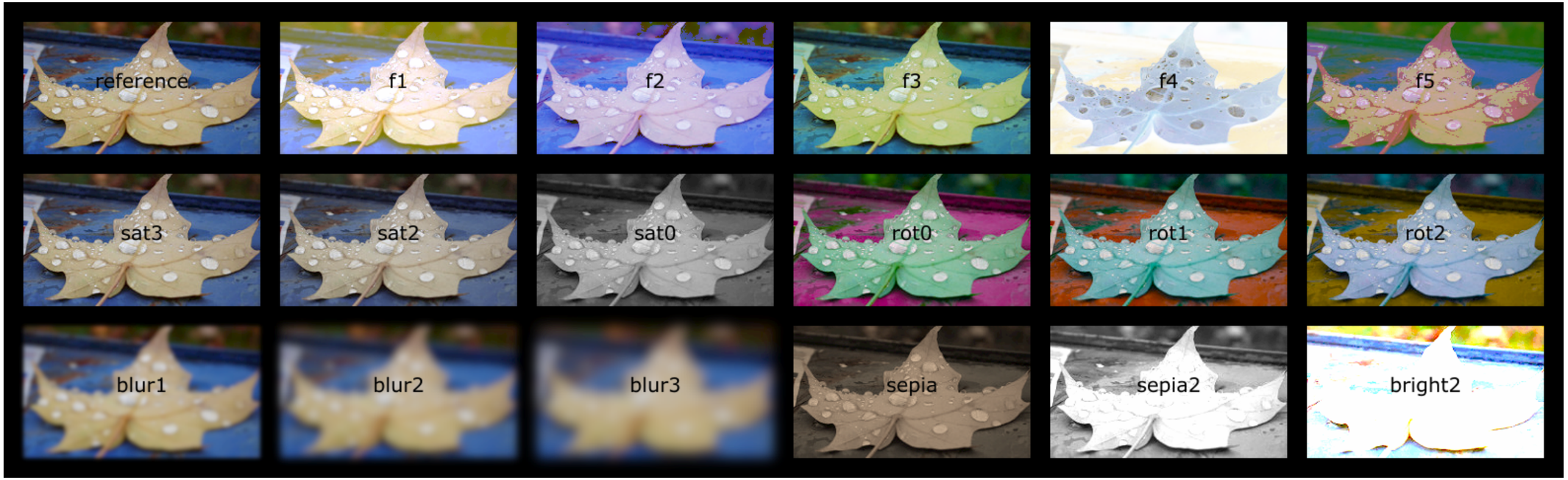
Textlines(x, y int, s []string, size, spacing int, fill, align string)

x, y w

h

Image(x, y, w, h int, link string, s ...string)

Filter Effects



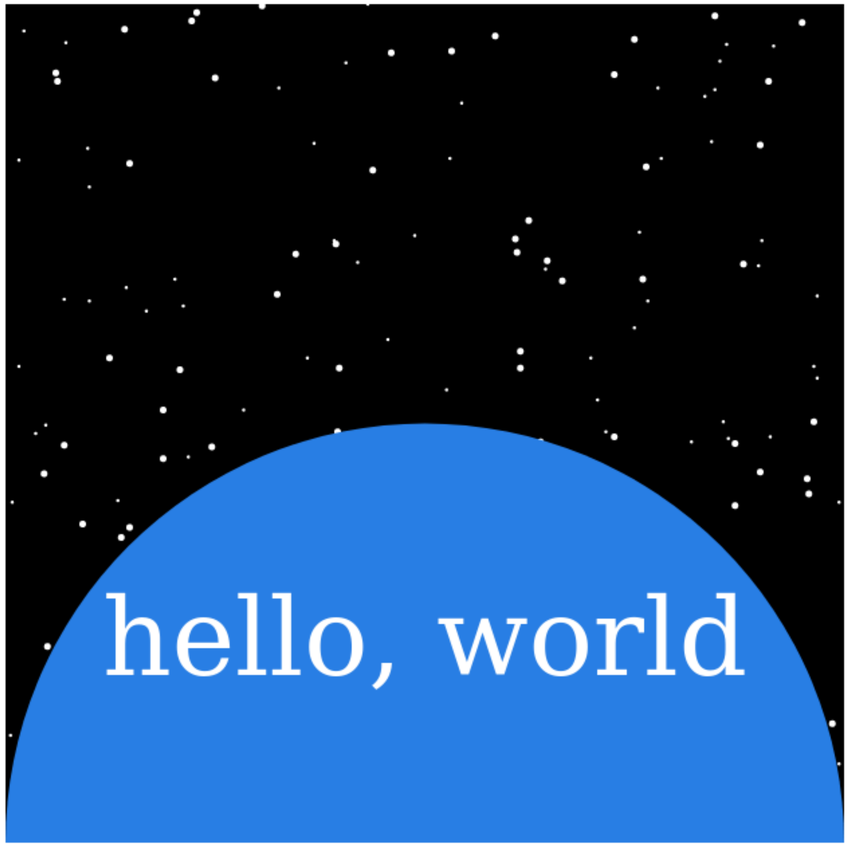
Sketching with code

svgplay: SVGo sketching

localhost:1999

Apps PROTOTYPE -events | Data... Other bookmarks

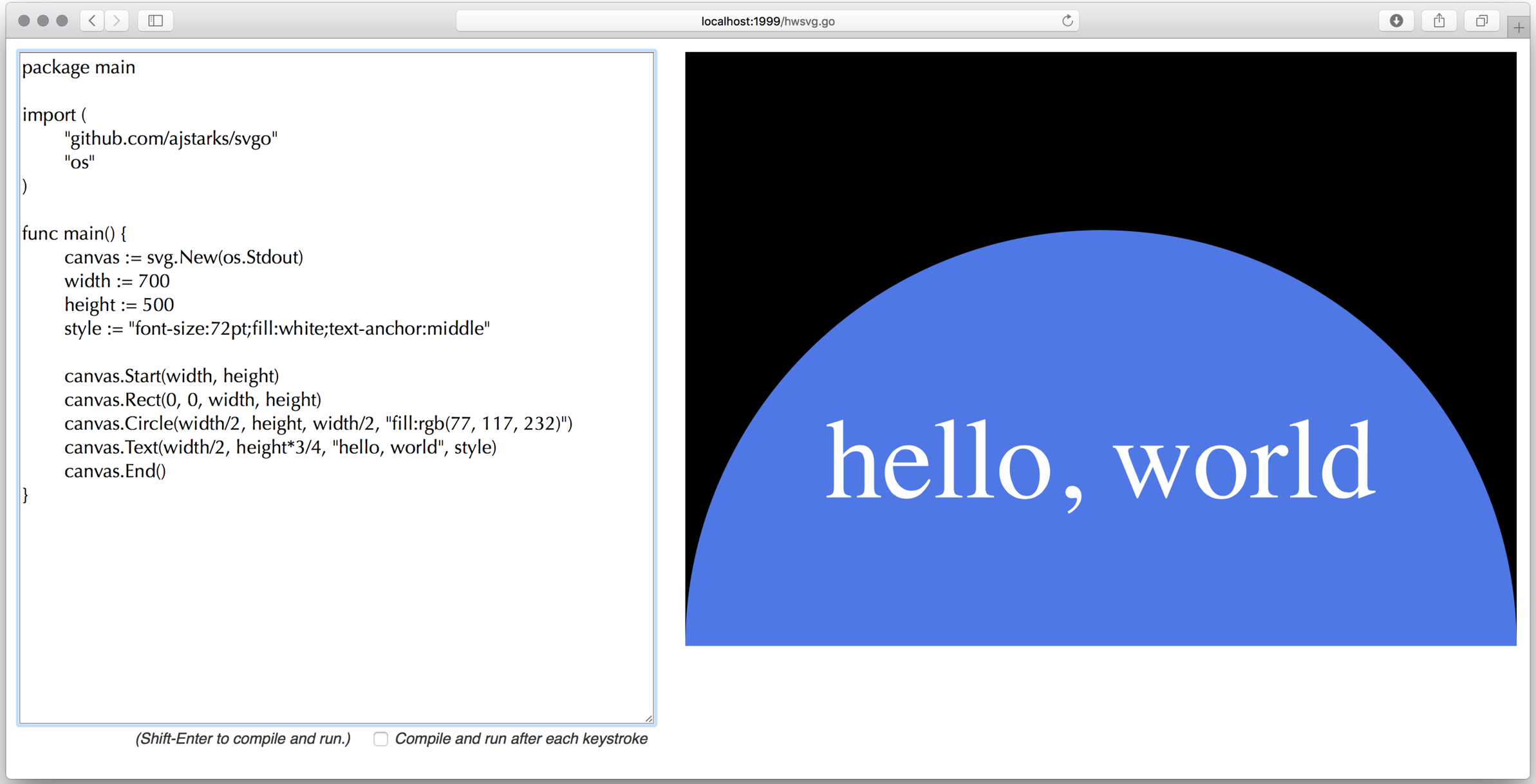
```
package main

import (
	"math/rand"
	"os"
	"time"
	"github.com/ajstarks/svgo"
)

func rn(n int) int { return rand.Intn(n) }

func main() {
	canvas := svg.New(os.Stdout)
	width := 500
	height := 500
	nstars := 250
	style := "font-size:48pt;fill:white;text-anchor:middle"

	rand.Seed(time.Now().Unix())
	canvas.Start(width, height)
	canvas.Rect(0,0,width,height)
	for i := 0; i < nstars; i++ {
		canvas.Circle(rn(width), rn(height), rn(3), "fill:white")
	}
	canvas.Circle(width/2, height, width/2, "fill:rgb(77, 117, 232)")
	canvas.Text(width/2, height*4/5, "hello, world", style)
	canvas.End()
}
```



(Shift-Enter to compile and run.) *Compile and run after each keystroke*

ajstarks@slab:~

```
$ svgplay
2019/06/18 19:19:44 ☠☠☠ Warning: this server allows a client connecting to 127.0.0.1:1999 to execute code on this computer ☠☠☠
```

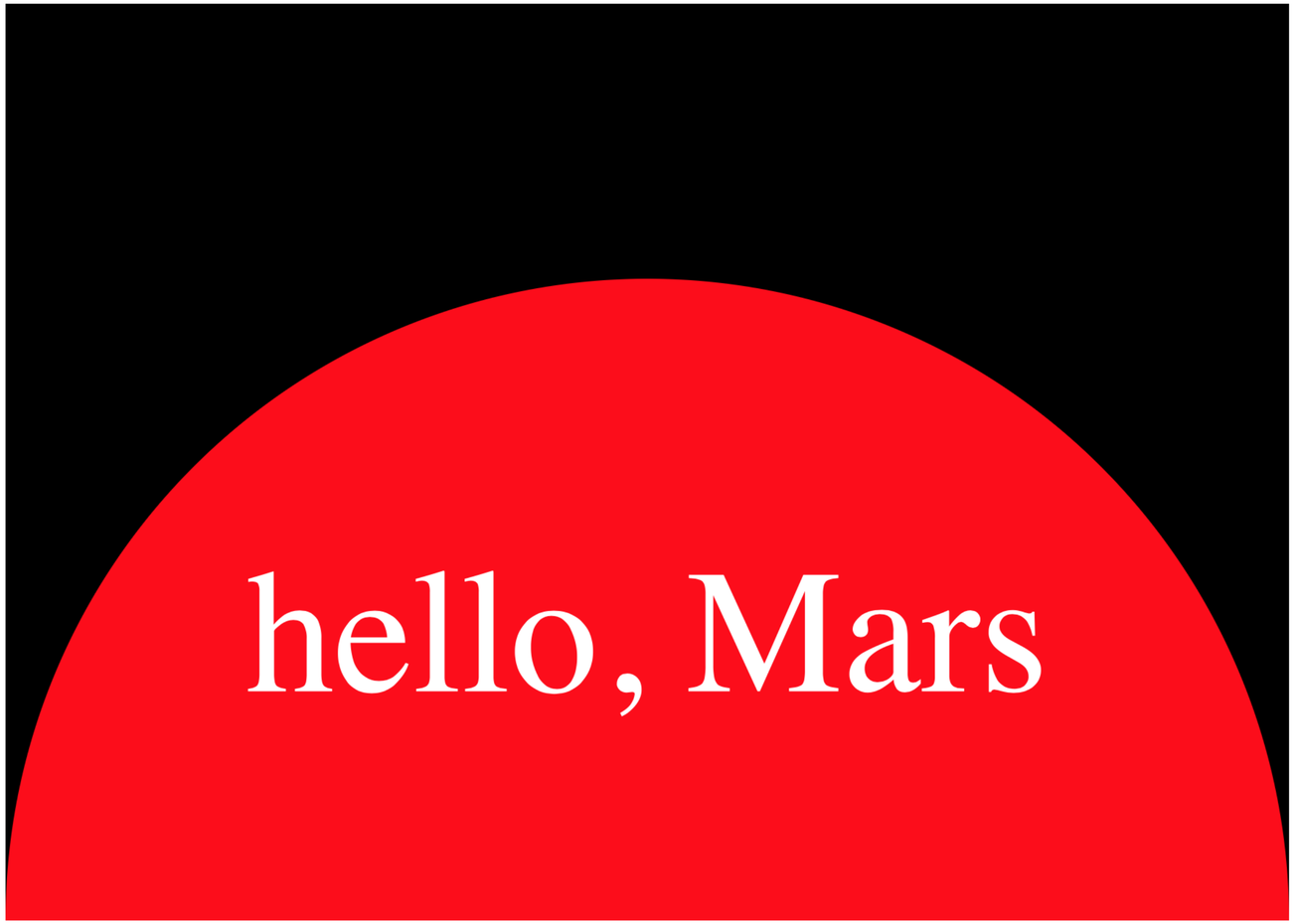

localhost:1999/hwsvg.go
package main
import (
"github.com/ajstarks/svgo"
"os"
)
func main() {
canvas := svg.New(os.Stdout)
width := 700
height := 500
style := "font-size:72pt;fill:white;text-anchor:middle"
canvas.Start(width, height)
canvas.Rect(0, 0, width, height)
canvas.Circle(width/2, height, width/2, "fill:rgb(77, 117, 232)")
canvas.Text(width/2, height*3/4, "hello, world", style)
canvas.End()
}
(Shift-Enter to compile and run.)
Compile and run after each keystroke
hello, world

svgplay

localhost:1999/hwsvg.go

```
package main

import (
	"github.com/ajstarks/svgo"
	"os"
)

func main() {
	canvas := svg.New(os.Stdout)
	width := 700
	height := 500
	style := "font-size:72pt;fill:white;text-anchor:middle"

	canvas.Start(width, height)
	canvas.Rect(0, 0, width, height)
	canvas.Circle(width/2, height, width/2, "fill:rgb(255,0,0)")
	canvas.Text(width/2, height*3/4, "hello, Mars", style)
	canvas.End()
}
```

(Shift-Enter to compile and run.) *Compile and run after each keystroke*



svgplay

Read/Parse/Draw Pattern

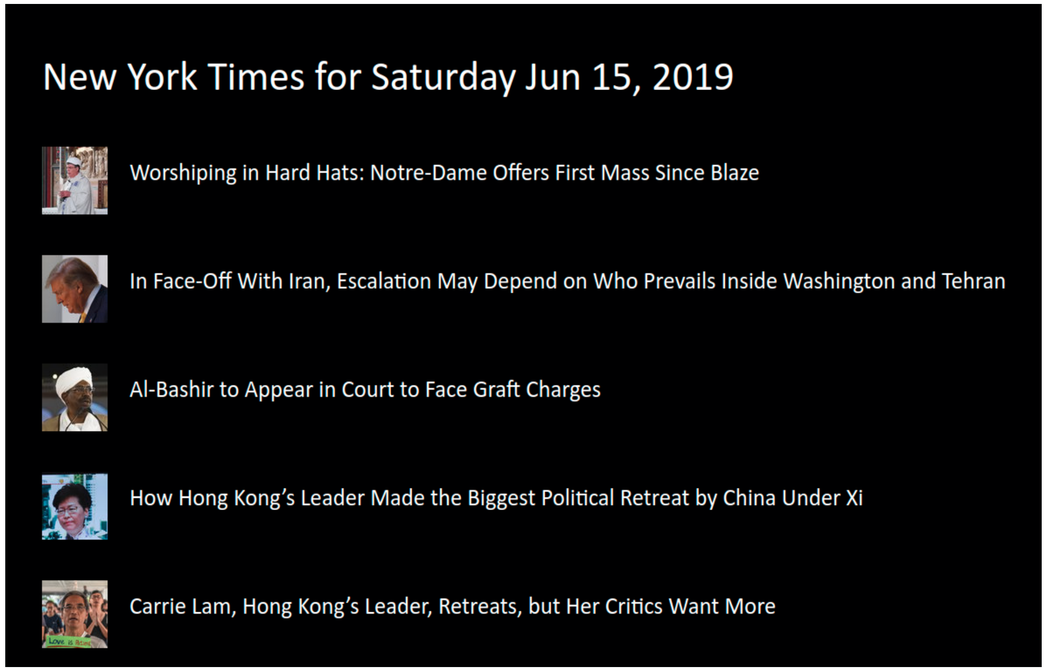
Read

Parse

Draw

https://api.nytimes.com/svc/news/v3/content/all/world/.json?api-key=...&limit=5


```
{
 "status": "OK",
 "copyright": "Copyright (c) 2019 The New York Times Company.  All Rights Reserved.",
 "num_results": 3313,
 "results": [
  {
   "slug_name": "15notredame",
   "section": "World",
   "subsection": "Europe",
   "title": "Worshiping in Hard Hats: Notre-Dame Offers First Mass Since Blaze",
   "abstract": "A small group celebrated their faith and a fragile cathedral, which remains closed amid rebuilding.",
   "url": "https://www.nytimes.com/2019/06/15/world/europe/notre-dame-mass.html",
   "byline": "By ADAM NOSSITER",
   "thumbnail_standard": "https://static01.nyt.com/images/2019/06/16/world/16notredame/16notredame-thumbStandard.jpg",
   "item_type": "Article",
   "source": "The New York Times",
   "updated_date": "2019-06-15T16:20:30-04:00",
   "created_date": "2019-06-15T16:04:47-04:00",
   "published_date": "2019-06-14T20:00:00-04:00",
   "first_published_date": "2019-06-15T16:03:17-04:00",
   "material_type_facet": "News",
  ...
  },
  {
   "slug_name": "15dc-iran",
   "section": "World",
   "subsection": "Middle East",
   "title": "In Face-Off With Iran, Escalation May Depend on Who Prevails Inside Washington and Tehran",
   "abstract": "The attacks in the Gulf of Oman emboldens the hard-liners in Iran and the U.S., each able to argue their longtime adversary is itching for war.",
   "url": "https://www.nytimes.com/2019/06/15/world/middleeast/trump-iran-hard-liners.html",
   "byline": "By DAVID E. SANGER and DAVID D. KIRKPATRICK",
   "thumbnail_standard": "https://static01.nyt.com/images/2019/06/16/us/16DC-IRAN/15dc-IRAN-thumbStandard.jpg",
   "item_type": "Article",
   "source": "The New York Times",
   "updated_date": "2019-06-15T15:48:01-04:00",
   "created_date": "2019-06-15T14:36:14-04:00",
   "published_date": "2019-06-14T20:00:00-04:00",
   "first_published_date": "2019-06-15T14:34:43-04:00",
   "material_type_facet": "News Analysis",
   "kicker": "News Analysis",
   "subheadline": null,
    }
   ]
  }
 ]
}
```

```
package main

import (
    "encoding/json"
    "flag"
    "fmt"
    "io"
    "net/http"
    "os"
    "time"

    svg "github.com/ajstarks/svgo"
)
```

Imports

```
// API Info and formats
const (
    NYTAPIkey = "7f35400618208b235360ee874ed70df0"
    NYTfmt    = "http://api.nytimes.com/svc/news/v3/content/all/%s/"
    param     = ".json?api-key=%s&limit=5"
    style     = "font-size:20pt;font-family:sans-serif;fill:white"
    errfmt    = "unable to get network data for %s (%s)"
    datefmt   = "Monday Jan 2, 2006"
    usage     = "(arts, health, sports, science, technology, u.s., world)"
)
```

Constants

```
// NYTHeadlines is the headline info from the New York Times
type NYTHeadlines struct {
    Status     string   `json:"status"`
    Copyright  string   `json:"copyright"`
    NumResults int      `json:"num_results"`
    Results    []result `json:"results"`
}

type result struct {
    Section    string `json:"section"`
    Subsection string `json:"subsection"`
    Title      string `json:"title"`
    Abstract   string `json:"abstract"`
    Thumbnail  string `json:"thumbnail_standard"`
}
```

Data Structures

```
func main() {
    var section = flag.String("h", "u.s.", usage)
    flag.Parse()
    canvas := svg.New(os.Stdout)
    canvas.Start(1200, 900)
    canvas.Rect(0, 0, 1200, 900)
    nytheadlines(canvas, *section)
    canvas.End()
}
```

Main

```
// netread derefernces a URL, returning the Reader, with an error
func netread(url string) (io.ReadCloser, error) {
    client := &http.Client{Timeout: 30 * time.Second}
    resp, err := client.Get(url)
    if err != nil {
        return nil, err
    }
    if resp.StatusCode != http.StatusOK {
        return nil, fmt.Errorf(errfmt, url, resp.Status)
    }
    return resp.Body, nil
}
```

Network

```
// nytheadlines retrieves data from the NYTimes API, decodes and displays it.
func nytheadlines(canvas *svg.SVG, section string) {
    // Read
    r, err := netread(fmt.Sprintf(NYTfmt+param, section, NYTAPIkey))
    if err != nil {
        fmt.Fprintf(os.Stderr, "headline read error: %v\n", err)
        return
    }
    defer r.Close()
    // Decode
    var data NYTHeadlines
    if err = json.NewDecoder(r).Decode(&data); err != nil {
        fmt.Fprintf(os.Stderr, "decode: %v\n", err)
        return
    }
    // drawing code next...
}
```

Read and Decode

```
// Positioning vars
thumbsize := 75
top, left := 200, 150
imx, titley := left - 100, top-100
x, y := left, top
ts := fmt.Sprintf("New York Times for %s", time.Now().Format(datefmt))

// Draw
canvas.Gstyle(style)
canvas.Text(imx, titley, ts, "font-size:175%")
for _, d := range data.Results {
    canvas.Text(x, y, d.Title)
    canvas.Image(imx, y-thumbsize/2, thumbsize, thumbsize, d.Thumbnail)
    y += 120
}
canvas.Gend()
```

Drawing

f.svg

File | /home/ajstarks/Decks/svgo-workshop/code/f.svg

New York Times for Wednesday Jun 19, 2019

Hong Kong Official Defends Police's Use of Force Against Protesters

Egypt Buries Morsi Quietly at Dawn

Could the Christchurch Attacks Have Been Prevented?

In U.K. Prime Minister Race, Boris Johnson Has Surprising Company

Canada Approves Expansion of Controversial Trans Mountain Pipeline

ajstarks@slab:~/Decks/svgo-workshop/code

```
$ go run nytsvg.go -h world  > f.svg
$
```