THE EVOLUTION of WordPress

Digital Innovation Conference
NYU, September 27, 2019

Youngjin Yoo

Weatherhead School of Management
Case Western Reserve University
xLAB

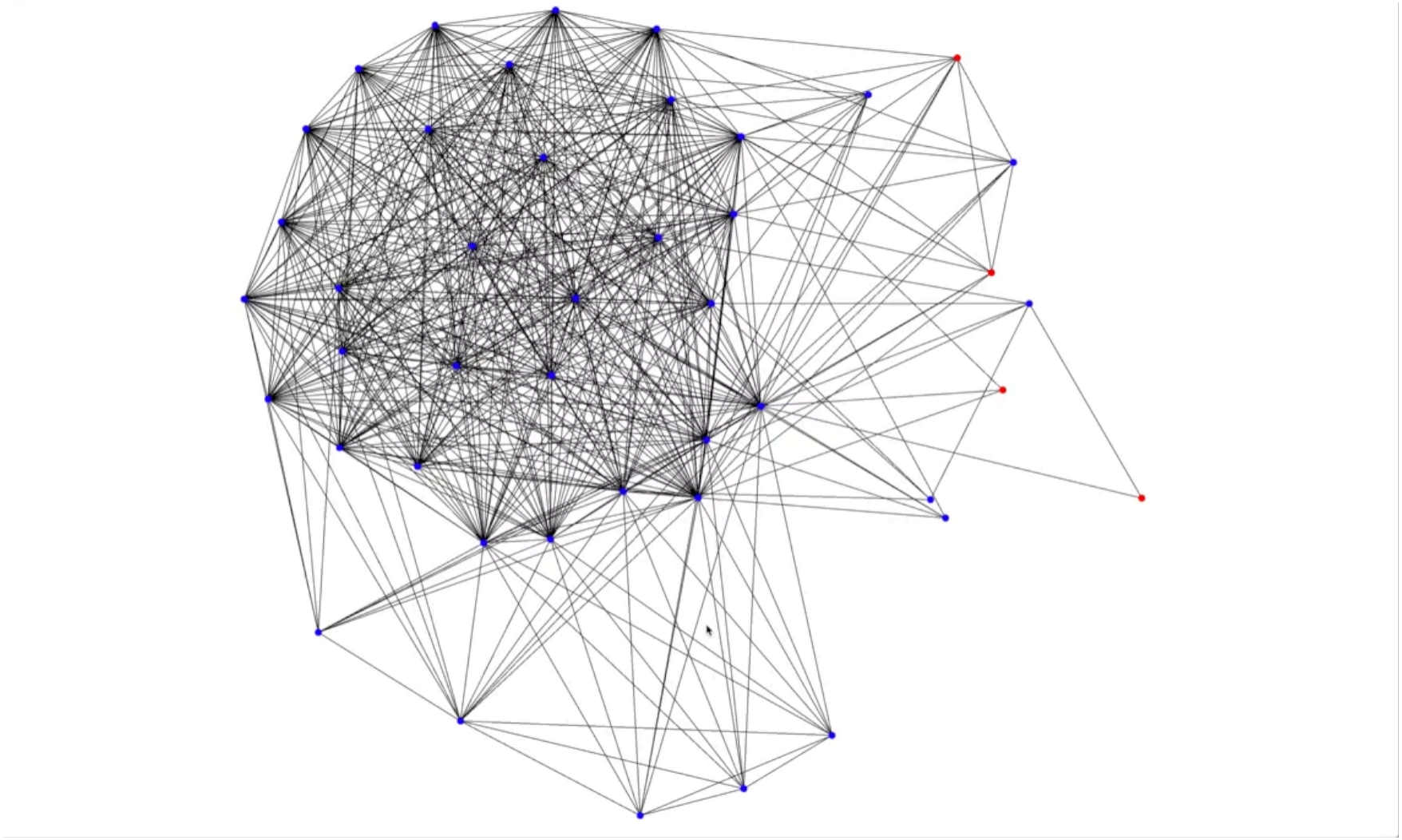
@YOUNGJINYOO

youngjin.yoo

GENERATIVITY Of DIGITAL PLATFORM ECOSYSTEM

1. How does a digital platform ecosystem evolve?

2. What are the possible roles that the platform owner can play in the evolutionary process?



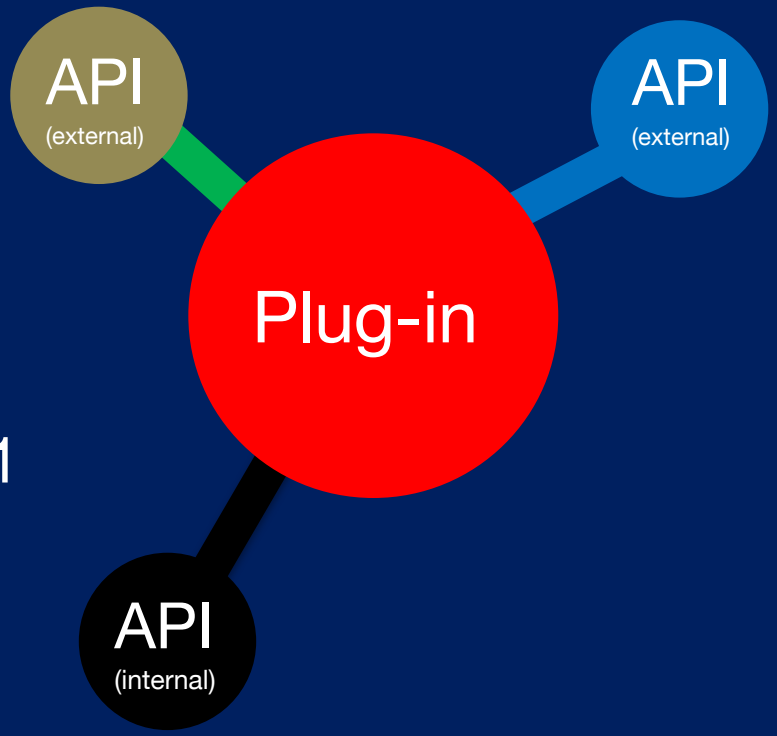

WordPress.org
English (UK)
Home
News
Get Started
Translations
Meet Ups
Events
Slack
Team Board
About Us: Our Mission
Get WordPress
Plugins
Extend your WordPress experience with 54,796 plugins.
Search plugins

Combinatorial Innovations

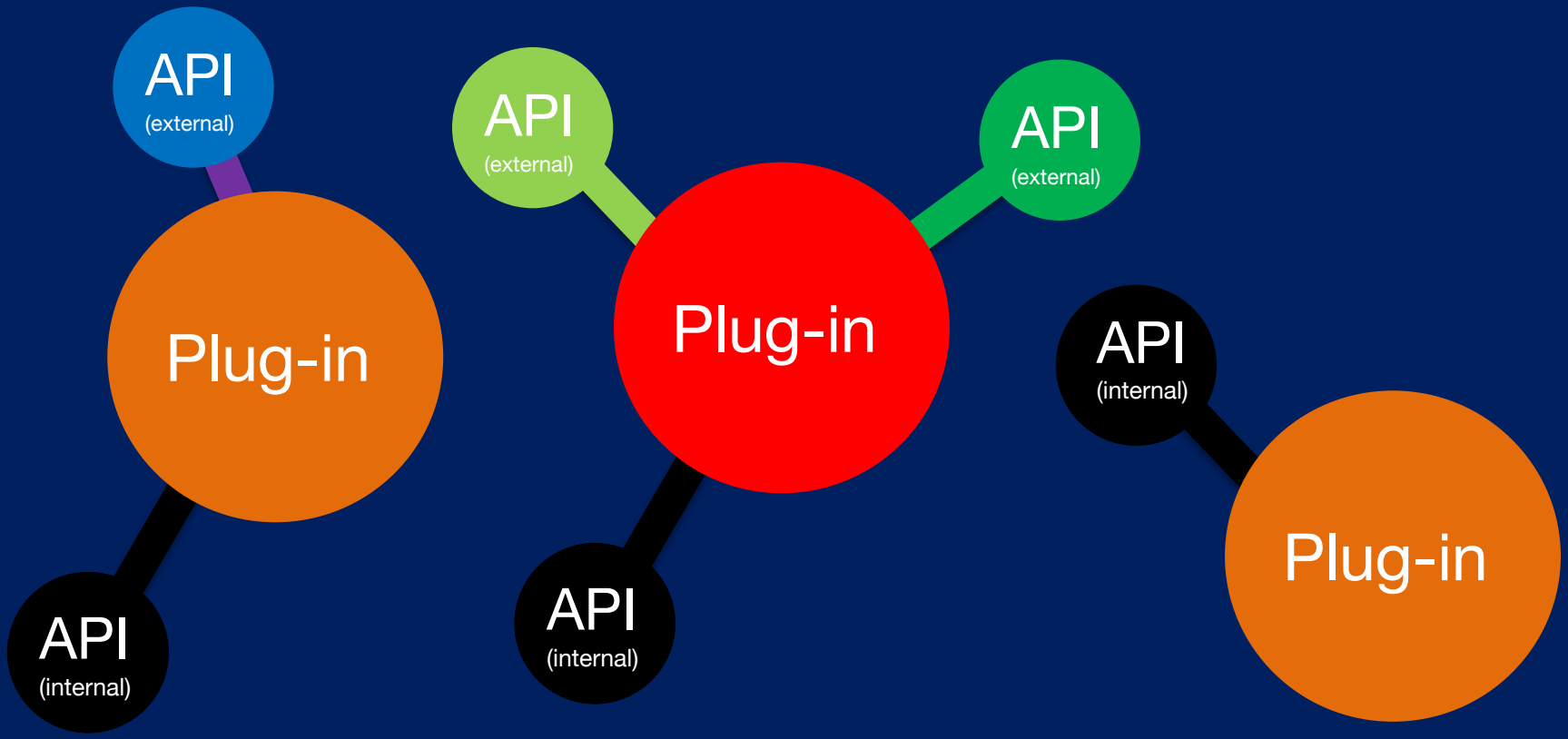
The combinatorial pattern of existing digital resources (APIs) used in complementary products produces the incredible generativity a digital platform ecosystem.

The WordPress Network

A hypothetical website with **one plug-in** that requires **three** Application Programming Interfaces (APIs), 2 external and 1 internal

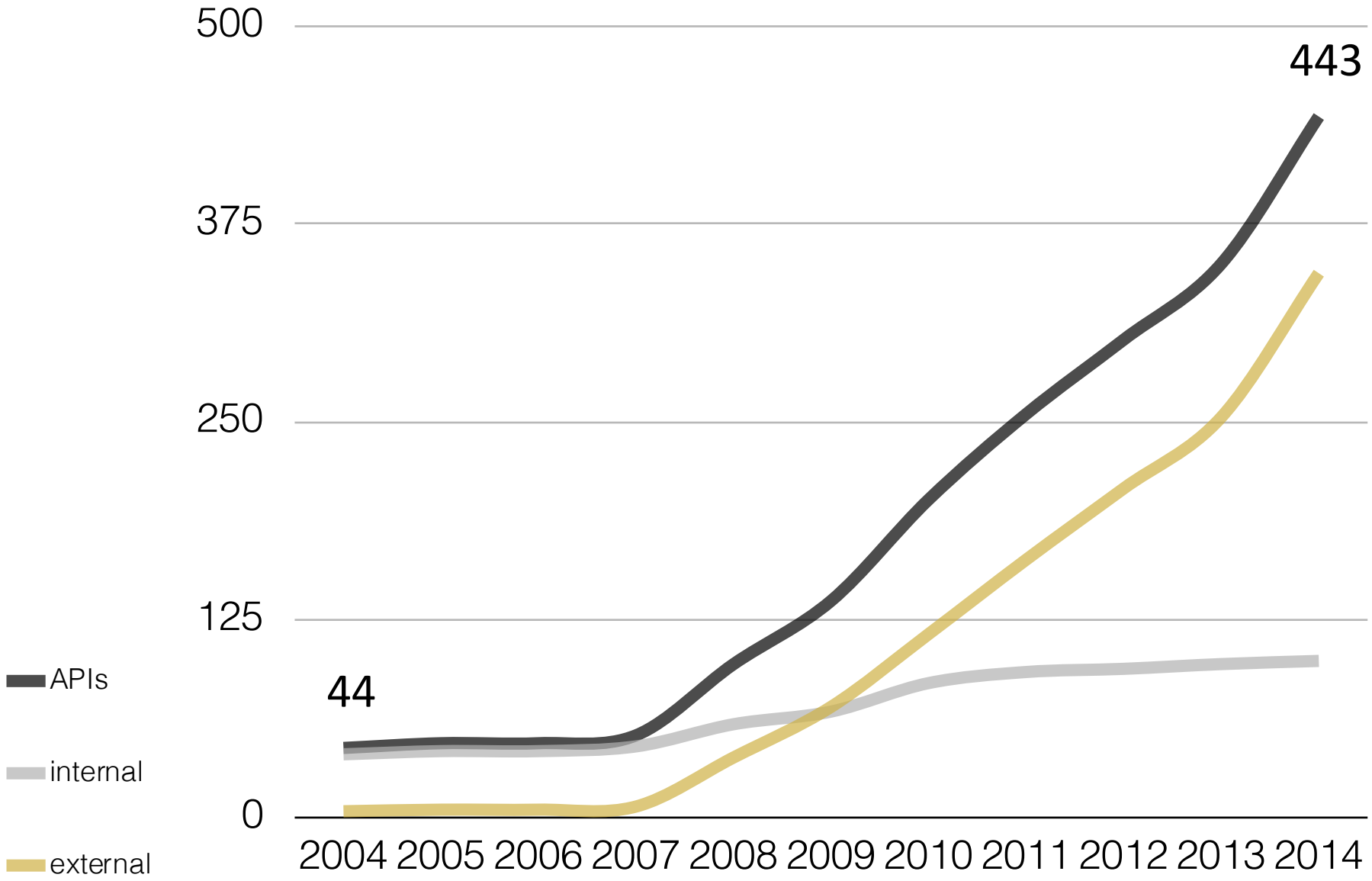


The WordPress Network

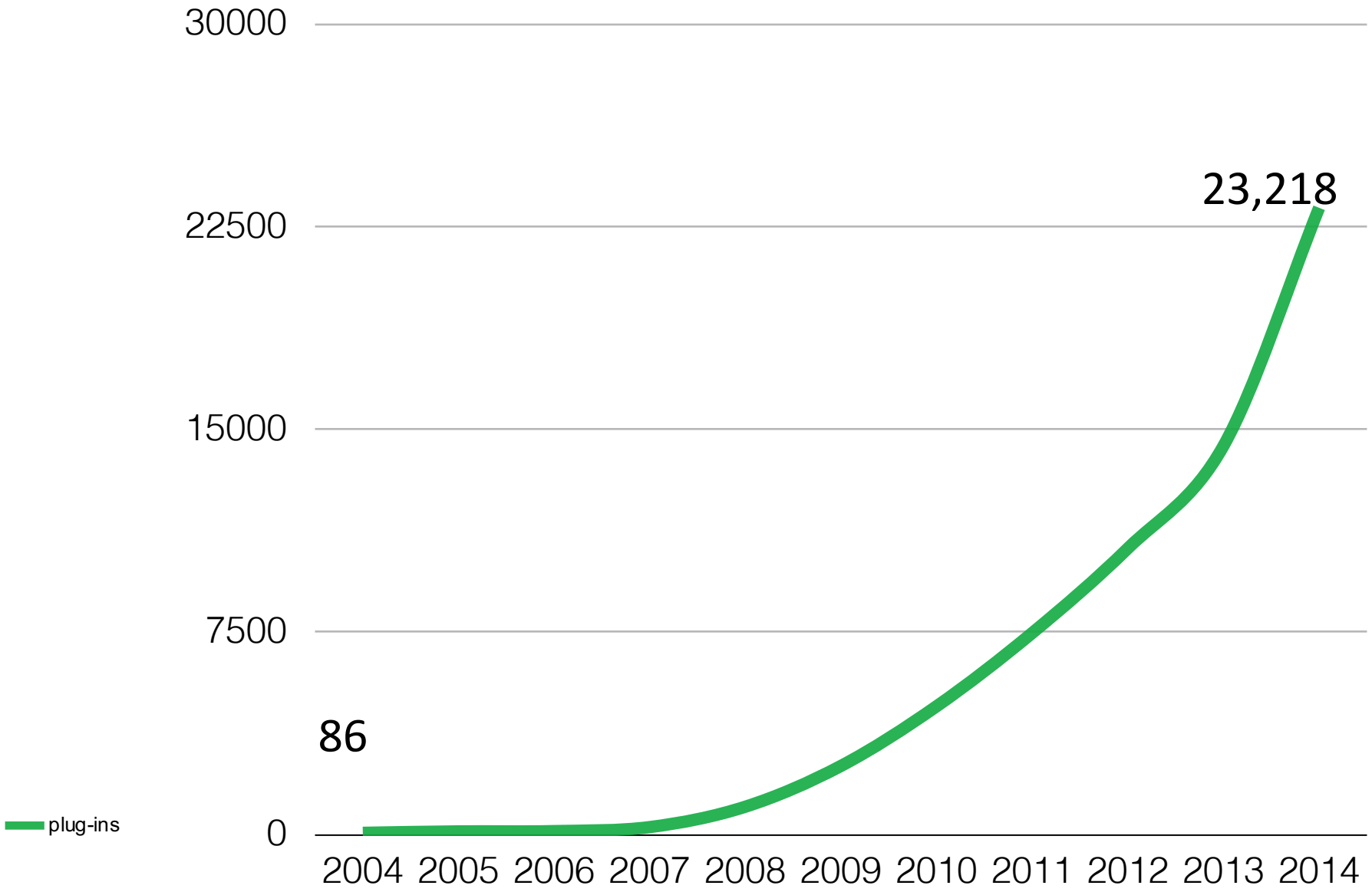


A hypothetical website with **three plug-ins** with 1to 3 Application Programming Interfaces (APIs).

Number of APIs



Number of Plug-ins



The Growth of WordPress

Year	# of plug-ins	Internal APIs	External APIs	Total APIs
2004	86	40	4	44
2005	139	42	5	47
2006	150	42	5	47
2007	298	45	7	52
2008	1052	59	38	97
2009	2562	67	70	137
2010	4770	85	116	201
2011	7483	92	163	255
2012	10615	94	208	302
2013	14409	97	253	350
2014	23218	99	344	443

Characterizing WordPress Ecosystem's Design Space

Design Space

23,218 x 443

23,218 different plug-ins through binary choices of 443 APIs

Variable	Description
$\overrightarrow{\boldsymbol{I}_{kt}}$	A vector of all internal APIs used in complementary products k at time t
$\overrightarrow{\boldsymbol{E}_{kt}}$	A vector of all external APIs used in complementary products k at time t
$\overrightarrow{\boldsymbol{P}_{kt}}$	A vector of all APIs used in complementary products k at time t
$\mathbf{I}_t$	A matrix of all internal APIs at time t
$\mathbf{E}_t$	A matrix of all external APIs at time t
$\mathbf{P}_t$	A matrix of all APIs in complementary products at time t
$\mathbf{S}_t$	A similarity matrix of complementary products at time t
$\mathbf{A}_t$	A weighted adjacency matrix of complementary products at time t
$\mathbf{W}_t$	A topological overlap matrix (TOM) of complementary products at time t
$\mathbf{\Gamma}_t$	A boundary resource (API) adjacency matrix at time t
p_{ij}	Element in $\mathbf{P}_t$, binary indicator of complementary product j using API i at time t
s_{ij}	The similarity measure between complementary products i and j in $\mathbf{S}_t$
a_{ij}	The adjacency indicator between complementary products i and j in $\mathbf{A}_t$
β	The soft thresholding number to suppress low correlations in a_{ij}
k_i	The sum of adjacency weights between complementary product i and neighbors
l_{ij}	The sum of adjacency weights when complementary products i and j are connected
w_{ij}	Element in TOM ($\mathbf{W}_t$), topological overlap measure between API i and j
γ_{ij}	Element in $\mathbf{\Gamma}_t$, a boundary resource (API) adjacency matrix at time t

$$\mathbf{P}_t = \begin{bmatrix} \mathbf{I}_t \\ \mathbf{E}_t \end{bmatrix} = \begin{bmatrix} I_{11} & I_{12} & \cdots & I_{1K} \\ I_{21} & I_{22} & \cdots & I_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ I_{M1} & I_{M2} & \cdots & I_{MK} \\ E_{11} & E_{12} & \cdots & E_{1K} \\ E_{21} & E_{22} & \cdots & E_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ E_{N1} & E_{N2} & \cdots & E_{NK} \end{bmatrix} = \begin{matrix} API_1 \\ API_2 \\ \vdots \\ API_H \end{matrix} \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1K} \\ p_{21} & p_{22} & \cdots & p_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ p_{H1} & p_{H2} & \cdots & p_{HK} \end{bmatrix}$$

$$\mathbf{S}_t = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1K} \\ s_{21} & s_{22} & \cdots & s_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ s_{K1} & s_{i2} & \cdots & s_{KK} \end{bmatrix}$$

where $s_{ij} = \dfrac{1 + corr\left(\overrightarrow{P_{it}}, \overrightarrow{P_{jt}}\right)}{2}$

$$\mathbf{A}_t = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1K} \\ a_{21} & a_{22} & \cdots & a_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ a_{K1} & a_{K2} & \cdots & a_{KK} \end{bmatrix}$$

where $a_{ij} = |s_{ij}|^{\beta}, \qquad \beta > 1$

$$\mathbf{W}_t = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1K} \\ w_{21} & w_{22} & \cdots & w_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ w_{K1} & w_{K2} & \cdots & w_{KK} \end{bmatrix}$$

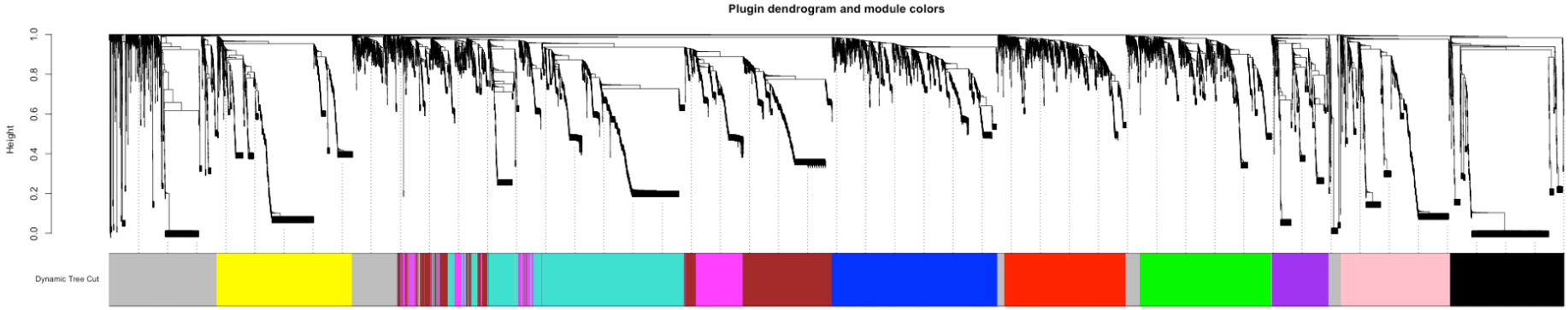
$$w_{ij} = \begin{cases} \dfrac{l_{ij} + a_{ij}}{\min(k_i, k_j) + 1 - a_{ij}} & if\ i \neq j \\ 1 & if\ i = j \end{cases}$$

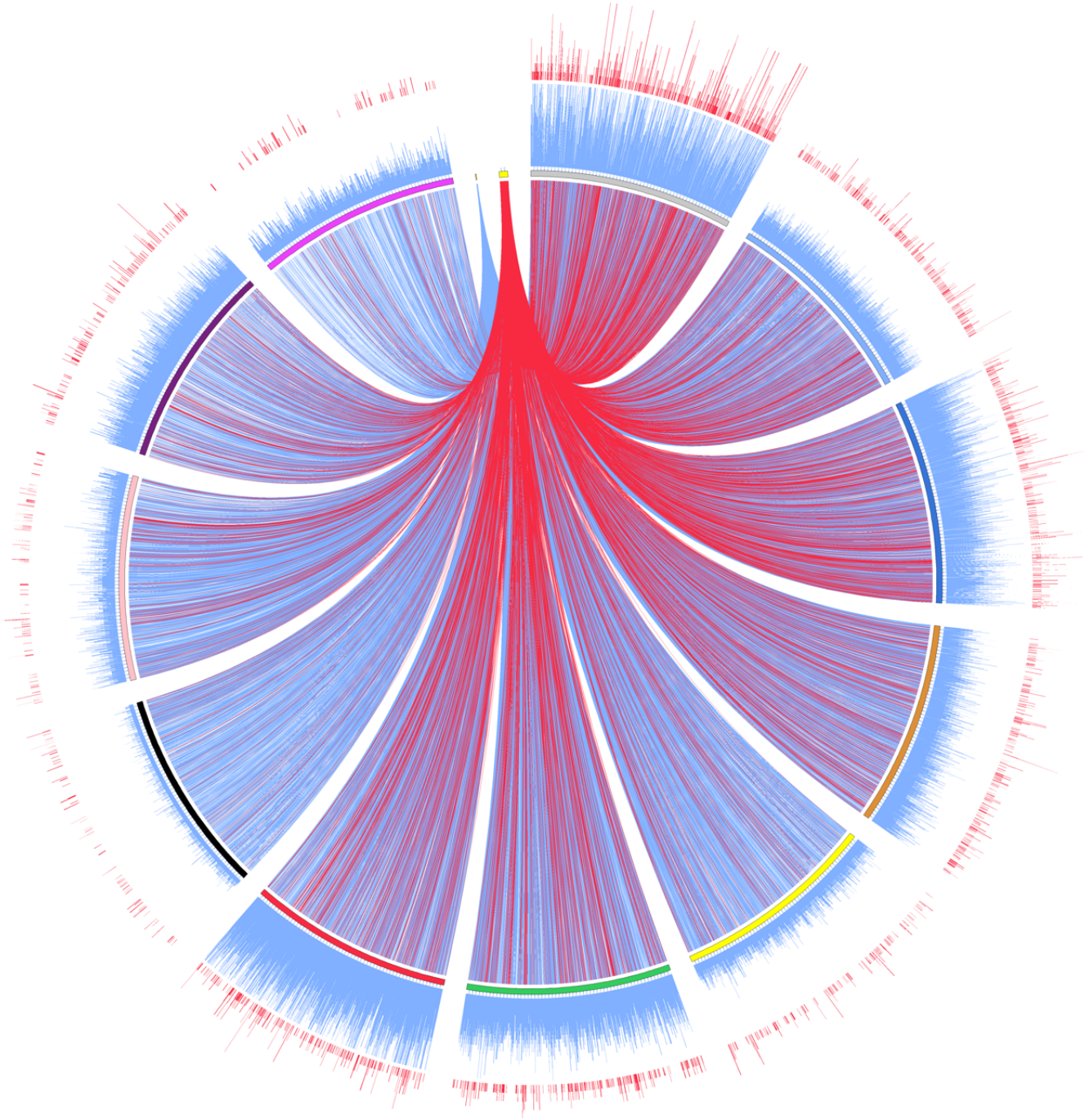
$$\mathbf{\Gamma}_t = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1H} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2H} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{H1} & \gamma_{H2} & \cdots & \gamma_{HH} \end{bmatrix}$$

Speciation

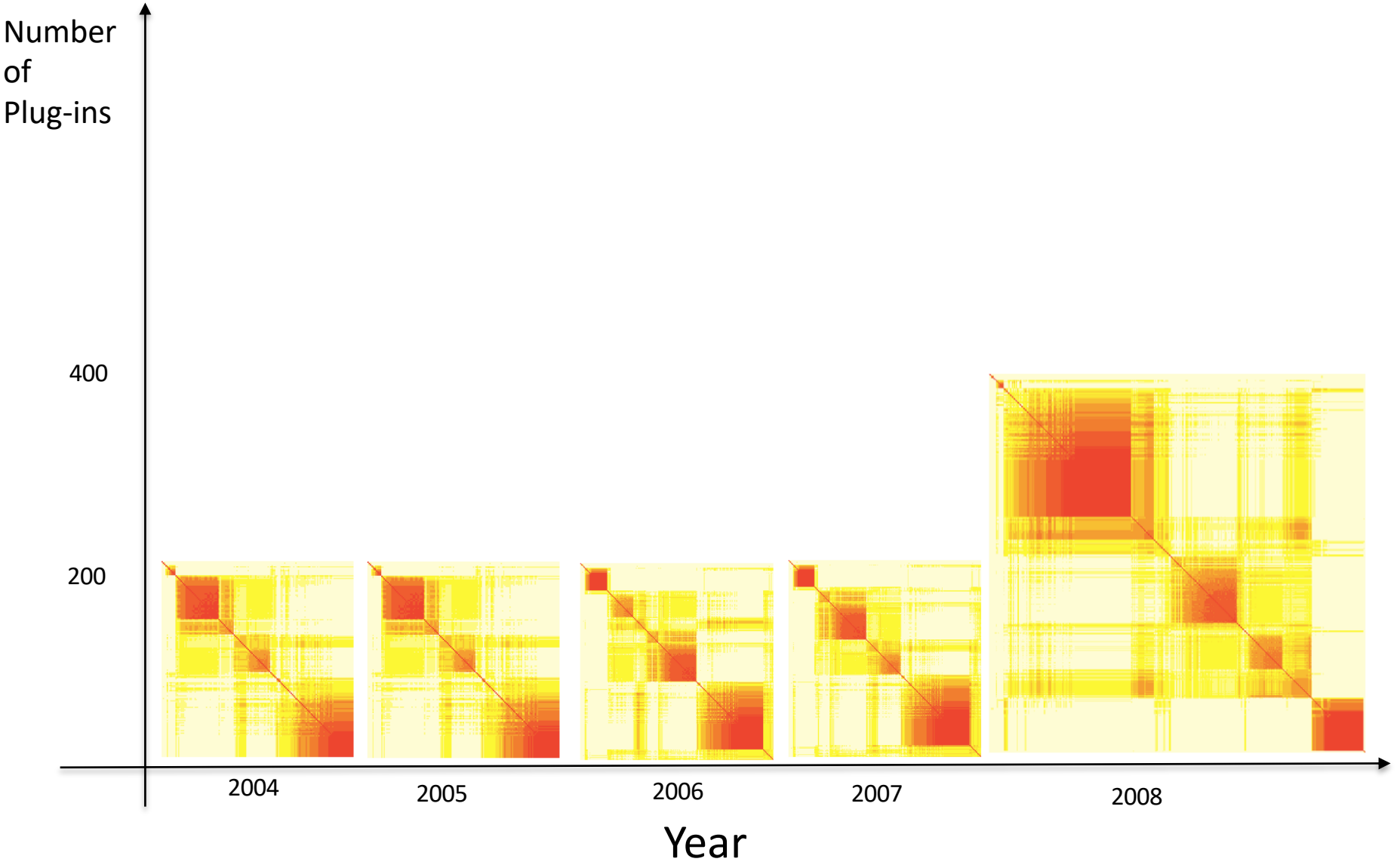
of WordPress Ecosystems

11 species of WordPress Plugs-ins

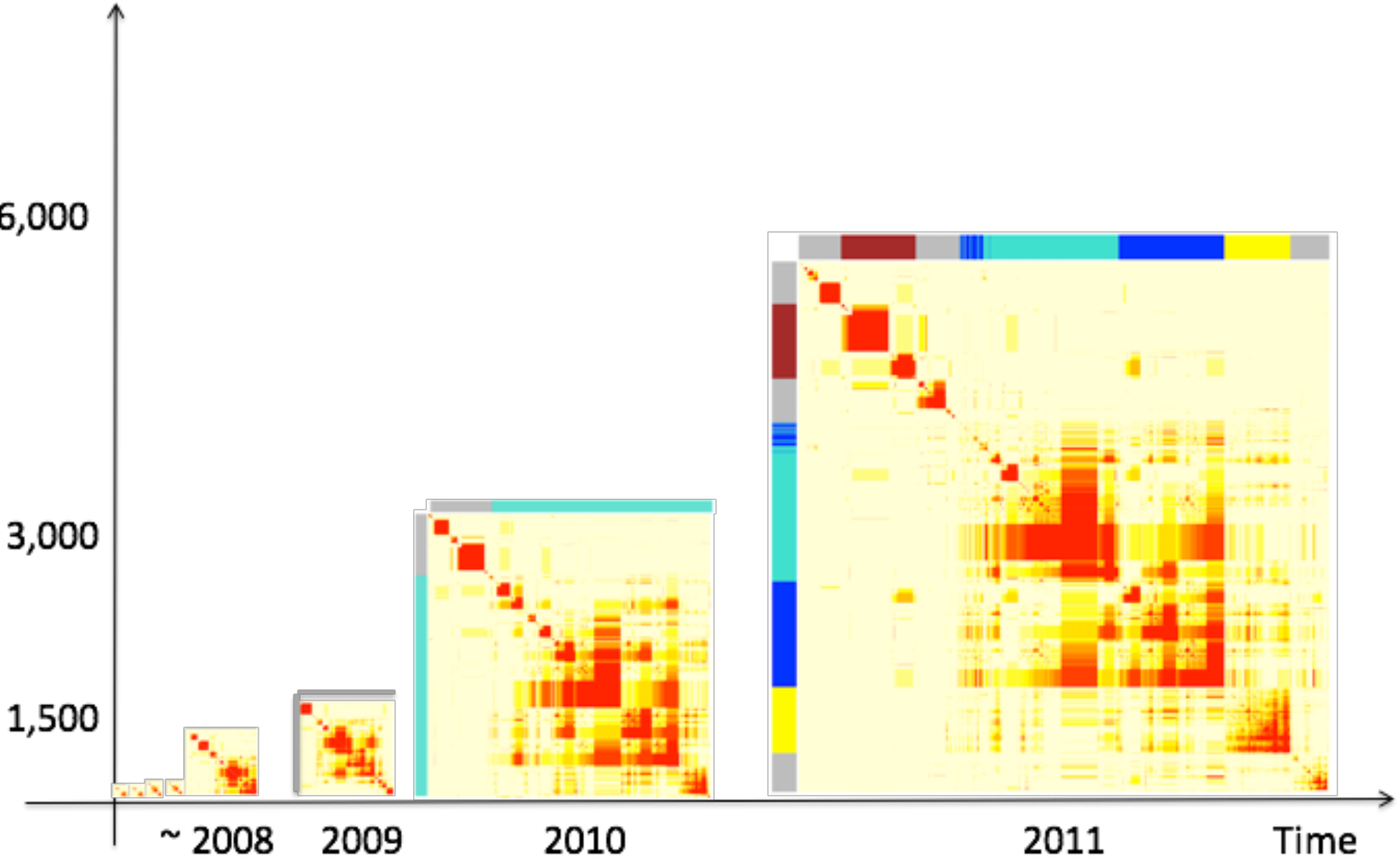




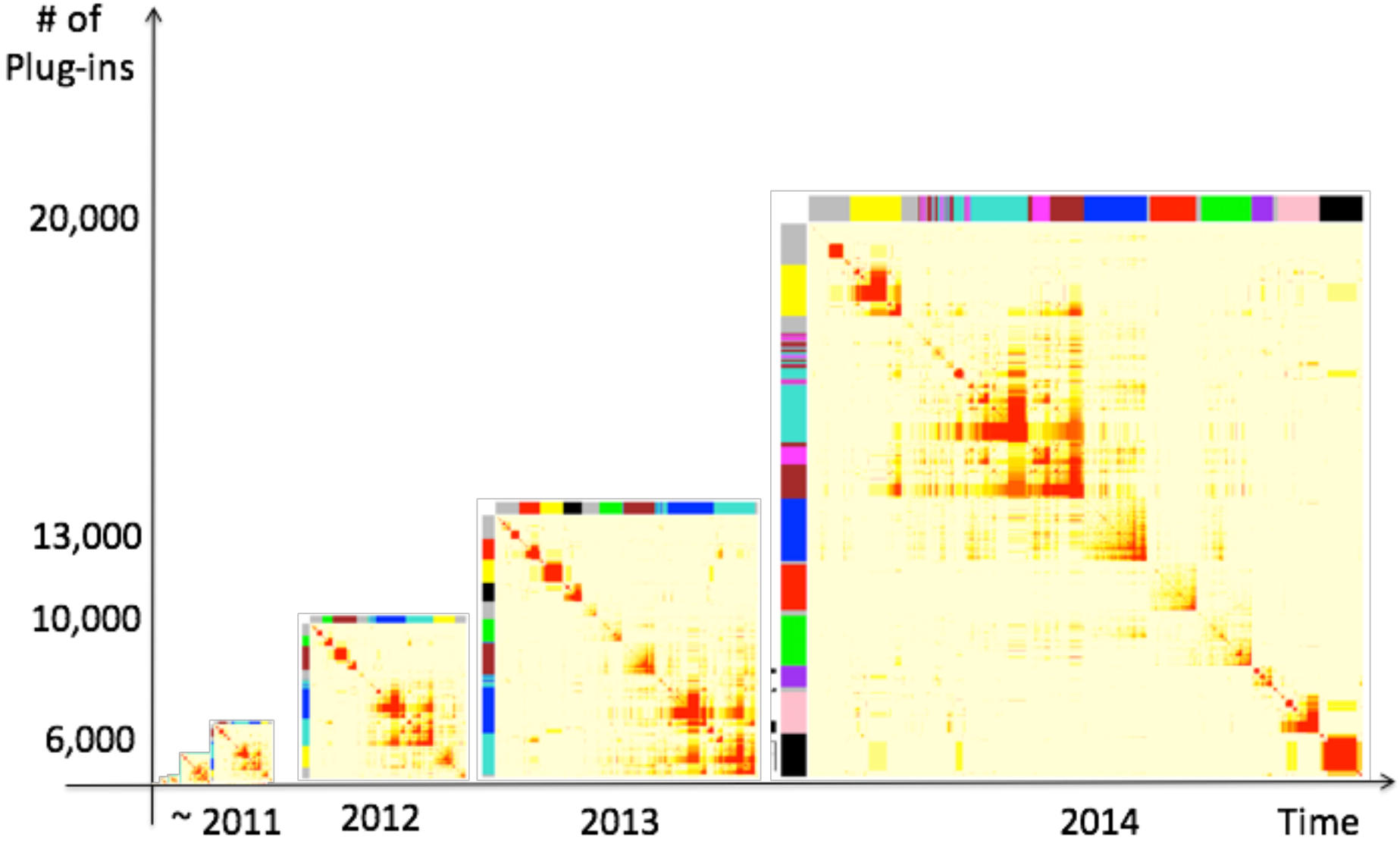
Plug-in Network (from 2004 to 2008)



Plug-in Network (from 2008 to 2011)



Plug-in Network (from 2011 to 2014)

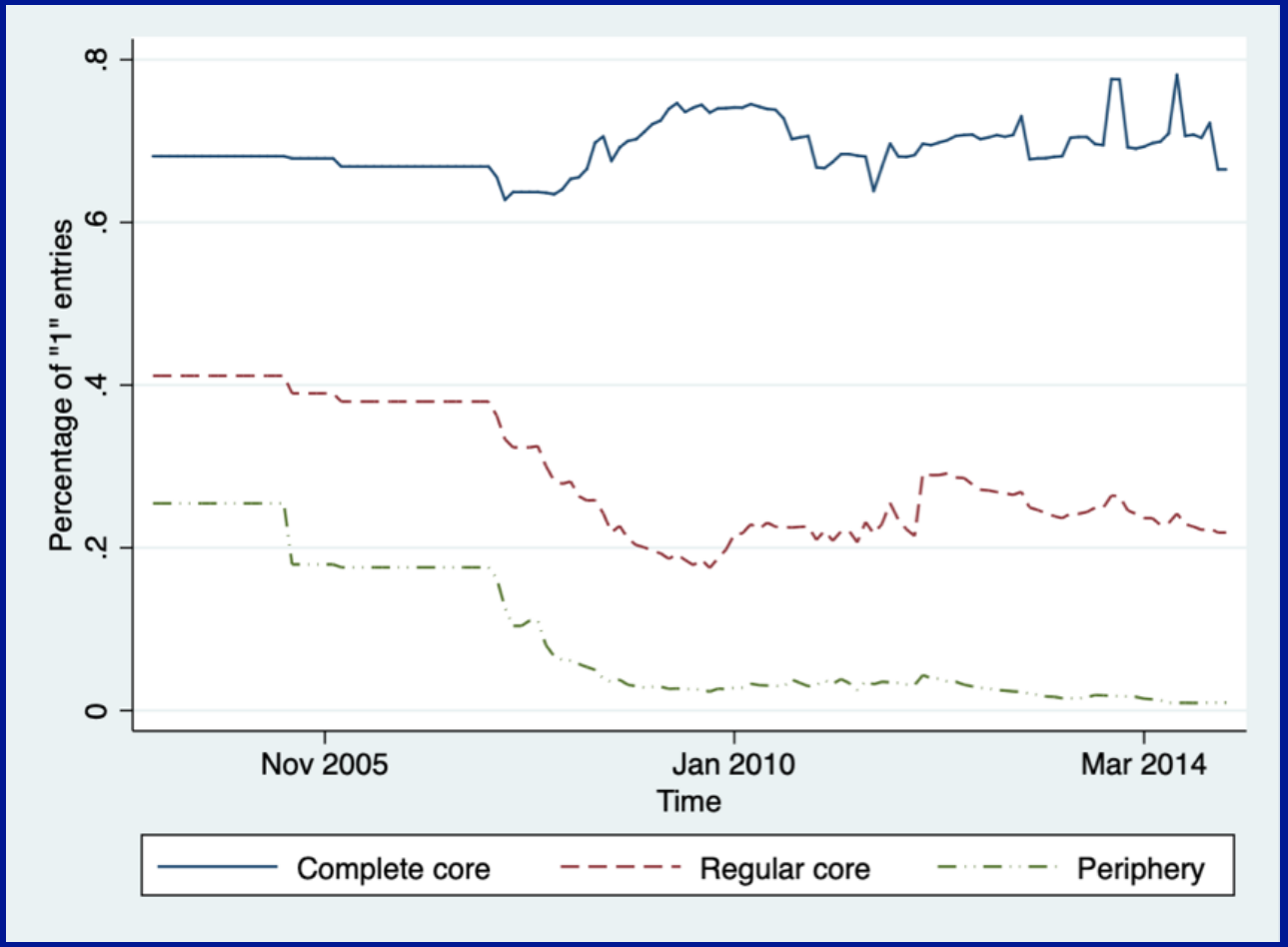


Co-Evolution

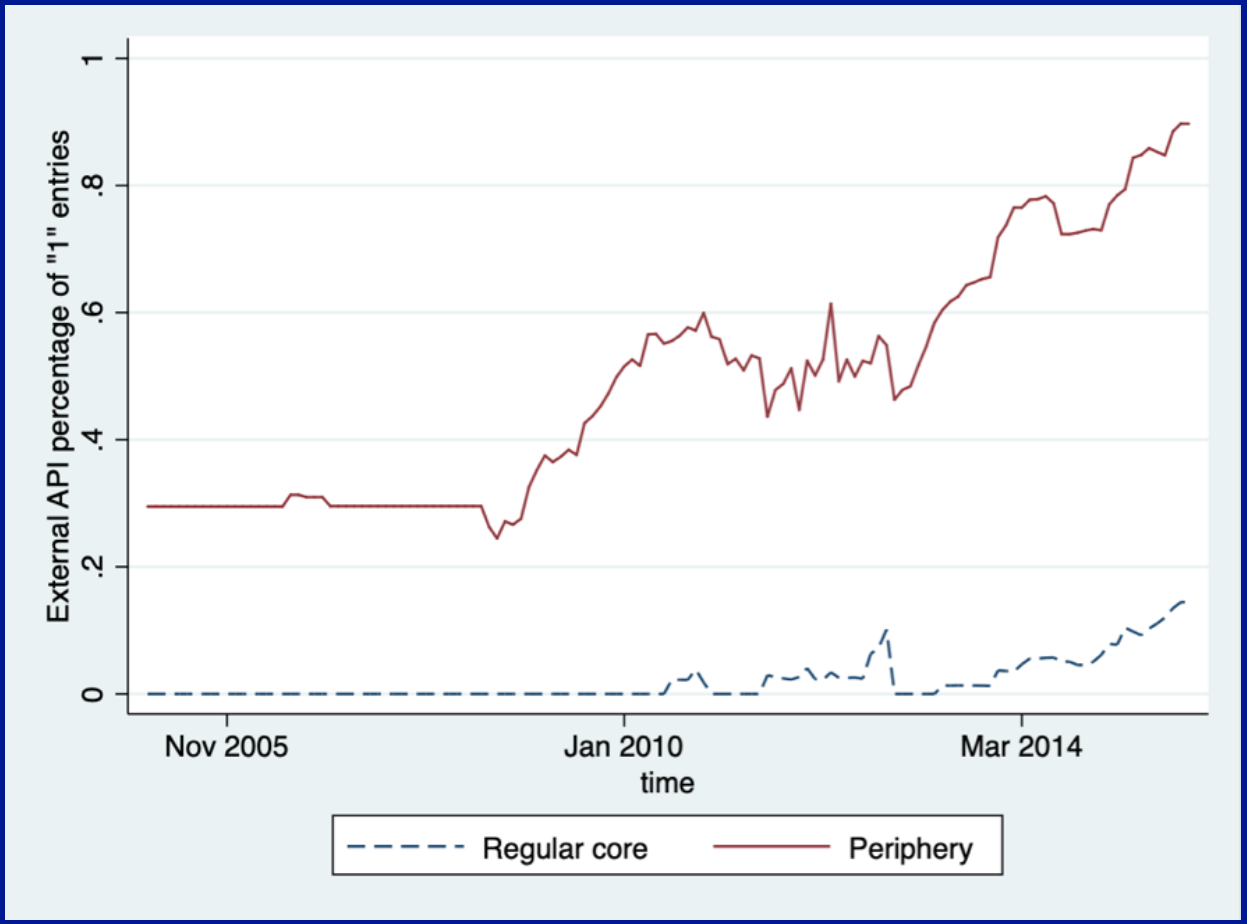
of WordPress Ecosystems

	total plug-in												total API	internal API	external API
2004	86	86	0	0	0	0	0	0	0	0	0	0	44	40	4
2005	139	139	0	0	0	0	0	0	0	0	0	0	47	42	5
2006	150	150	0	0	0	0	0	0	0	0	0	0	47	42	5
2007	298	298	0	0	0	0	0	0	0	0	0	0	52	45	7
2008	1052	328	724	0	0	0	0	0	0	0	0	0	97	59	38
2009	2562	564	1998	0	0	0	0	0	0	0	0	0	137	67	70
2010	4770	500	1860	1485	925	0	0	0	0	0	0	0	201	85	116
2011	7483	1775	1961	1752	1053	942	0	0	0	0	0	0	255	92	163
2012	10615	2304	2297	2179	1632	1507	696	0	0	0	0	0	302	94	208
2013	14409	2473	2796	2720	1730	1270	1248	1140	1032	0	0	0	350	97	253
2014	23218	3080	3257	2561	2319	2161	2102	1934	1808	1737	1272	897	443	99	344

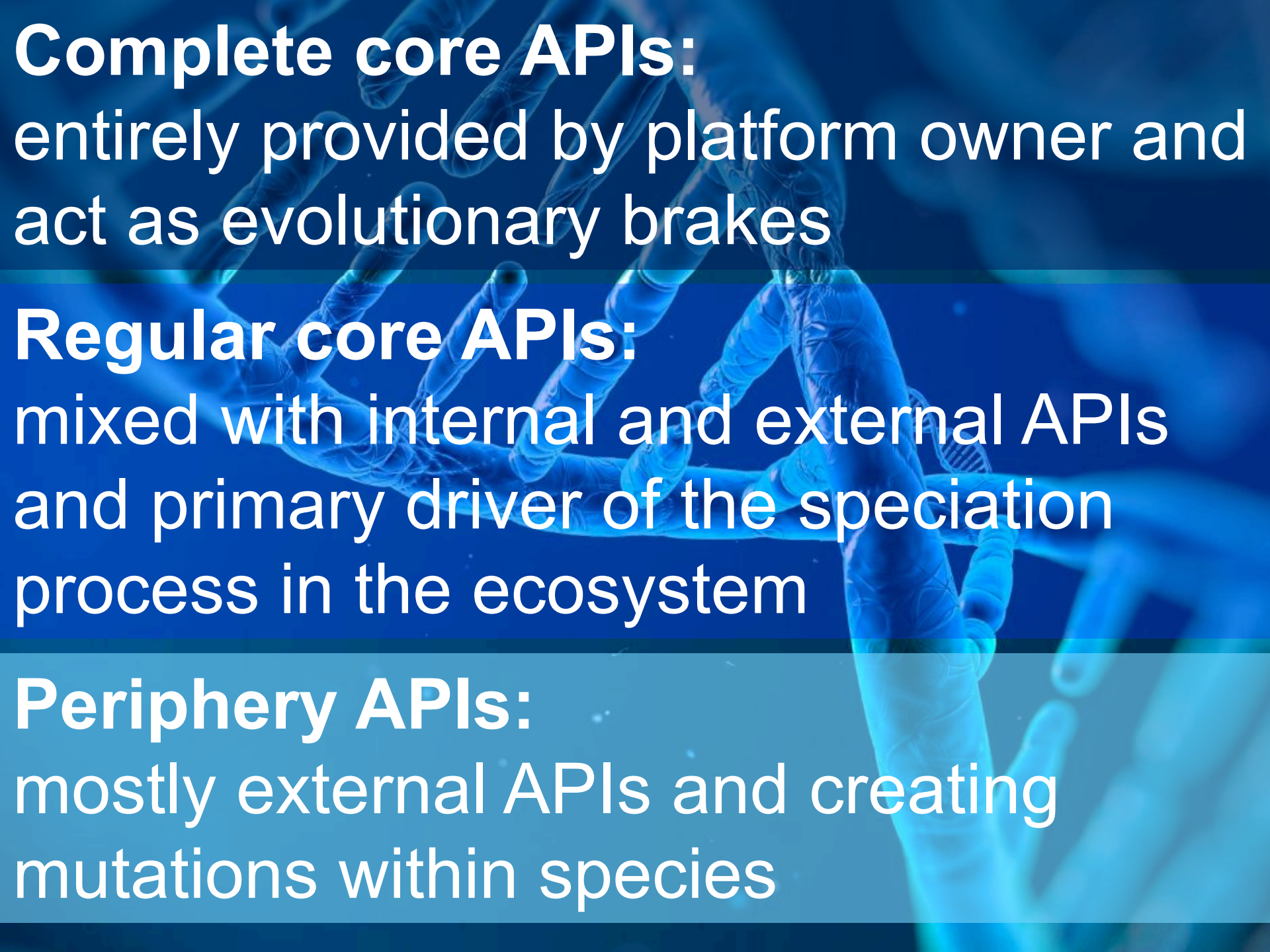

Genes
in WordPress Ecosystems



3 different types of genes (APIs) appear



Where do **regular core genes** come from?

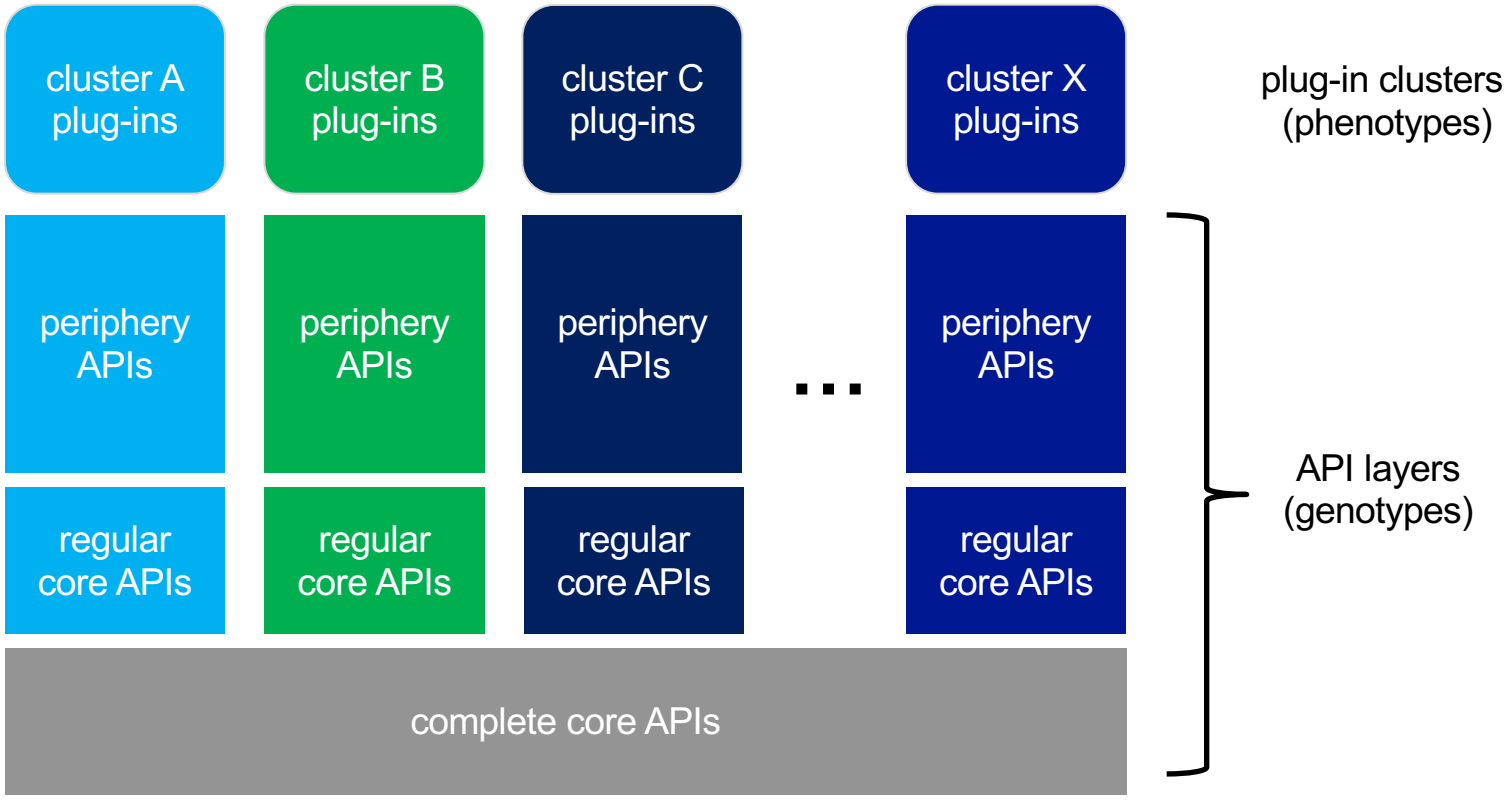


Complete core APIs:
entirely provided by platform owner and act as evolutionary brakes

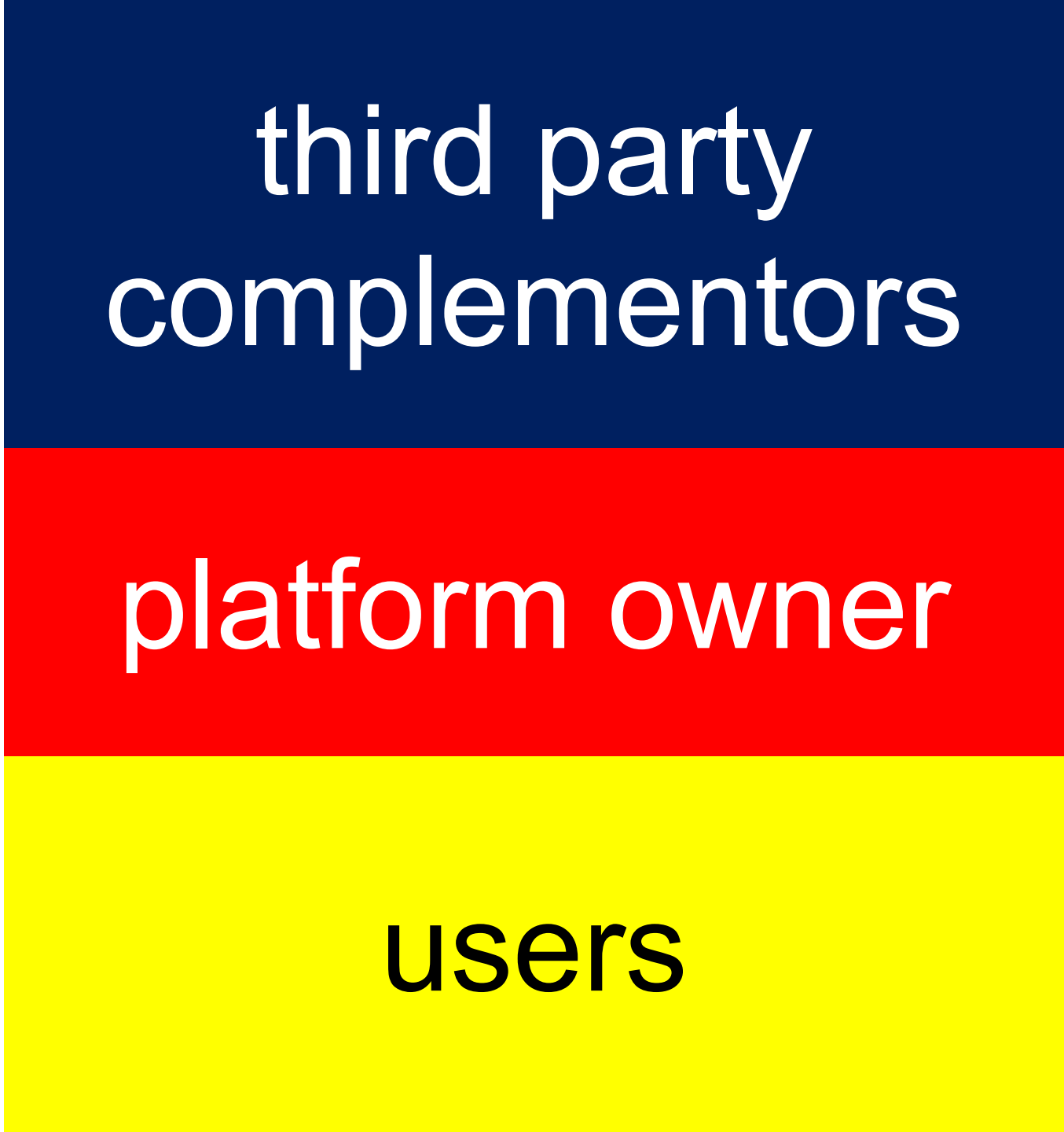
Regular core APIs:
mixed with internal and external APIs and primary driver of the speciation process in the ecosystem

Periphery APIs:
mostly external APIs and creating mutations within species

Generative Structure
of WordPress Ecosystems



Existing View



Our View

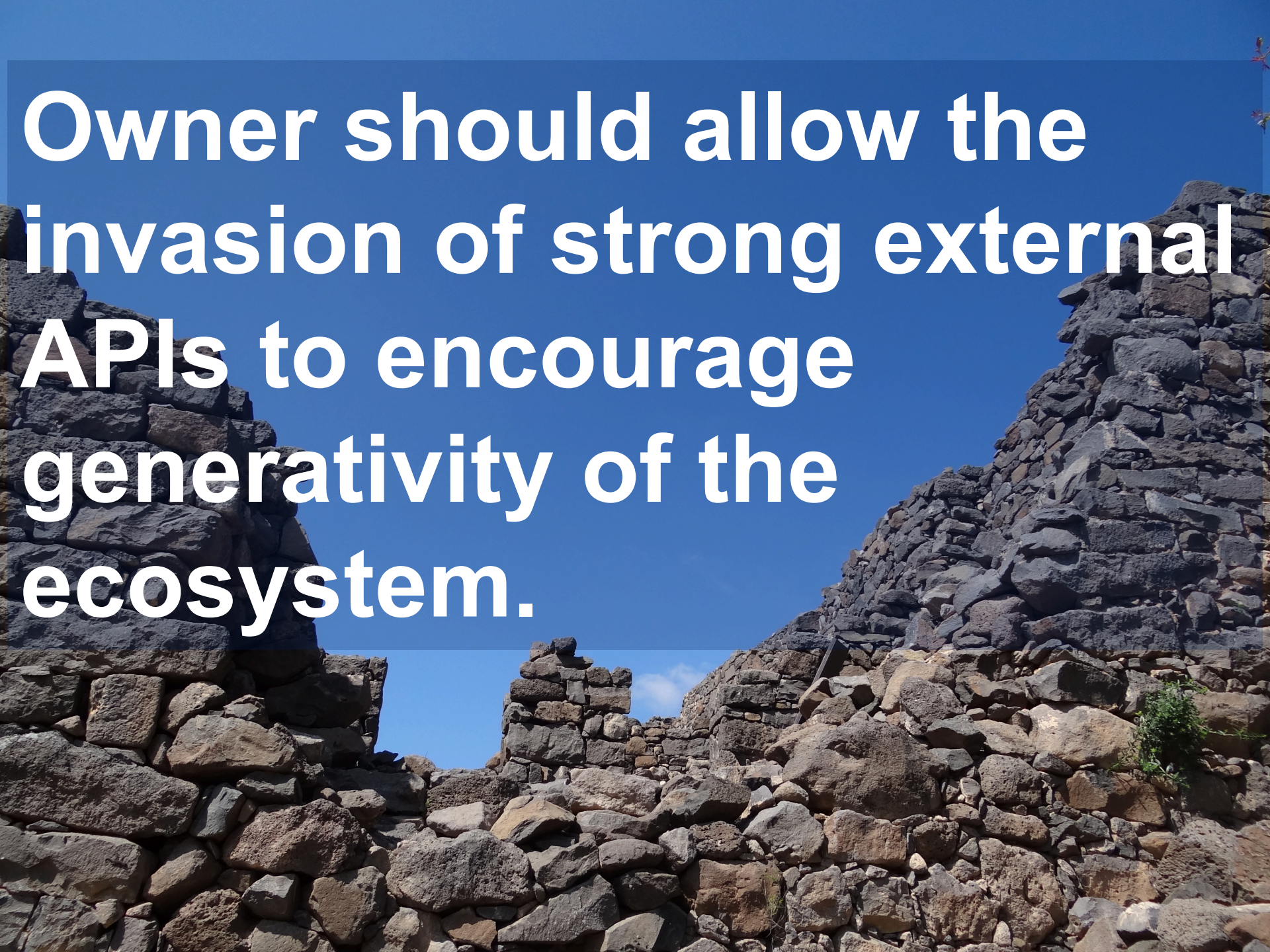
third party complementors

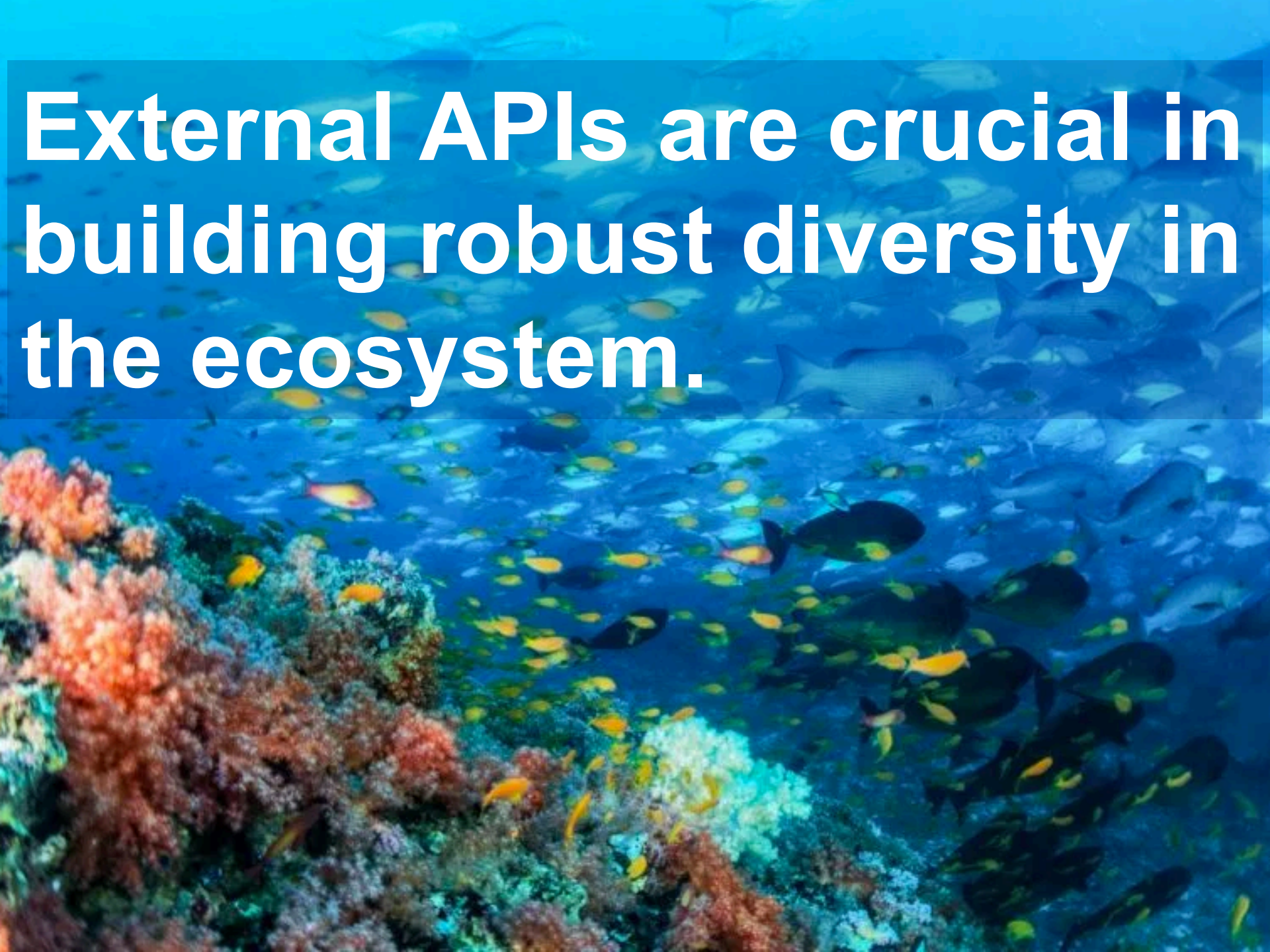
API Builders

platform owner

users

Owner's APIs set the boundary of the ecosystems


Owner should allow the invasion of strong external APIs to encourage generativity of the ecosystem.


External APIs are crucial in building robust diversity in the ecosystem.